JOB - SHOP SCHEDULING

BY

OM PRAKASH YADAV

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

SEPTEMBER 1972

JOB - SHOP SCHEDULING

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

BY

OM PRAKASH YADAV

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

SEPTEMBER 1972

CERTIFICATE

This is to certify that the work "Job - Shop Scheduling" has been carried out under my supervision and has not been submitted elsewhere for a degree.

(Dr. Z. Mehdi)
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology Kanpur

ACKNOWLEDGEMENT

I am greatly indebted to Dr. Z. Mehdi for his help and guidance in the compilation of this work. I am deeply grateful to Dr. J.L. Batra for his useful suggestions and discussions.

I thank all my friends who extended every cooperation in bringing out the final copy of this manuscript. Lastly, I thank Mr. J.D. Varma who patiently typed the manuscript.

Om Prakash Yadav

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE NO. PAGE

NOMENCLATURE

t_{ij}	=	Processing time for the j^{th} operation of the i^{th} job.
d_i	=	Due date of the i^{th} job.
c_i	=	Number of identical machines in the i^{th} machine centre.
a_i	=	Processing time of the i^{th} job for a single machine problem.
N_i	=	Number of operations required to complete the i^{th} job.
J_i	=	Present number of the operation of the i^{th} job.
s_{ij}	=	Set-up time required for the j^{th} job when the preceding job is the i^{th} job.
n_{ij}	=	Efficiency of the i^{th} operator on the j^{th} machine.
$S_{t,1}$	=	Time at which the first job is ready for its first operation.
σ	=	Standard deviation.
FDFS	=	First Demand First Served.
MAXQ	=	Maximum Queue Length.
SST	=	Shortest Service Time.
DUDT	=	Critical Due - Date.
MINSEQ	=	Minimum Set-up Time Sequence.
FIXSETSEQ	=	Fixed Set-up Sequence.
SPT	=	Shortest Process Time.
LST	=	Largest Service Time.

SYNOPSIS

Om Prakash Yadav, M.Tech. (Mech.), Indian Institute of Technology, Kanpur, September, 1972, "Job - Shop Scheduling".

A simulation experiment has been designed to test the performance of different combinations of job assignment and labor assignment rules for a job shop. The performance of Job Assignment Rules and Labor Assignment Rules have been tested with respect to the criteria of minimizing Flow - Time, Job-Lateness, Idle - Time, Job - Tardiness and number of jobs completed later than their due - dates. Effects of varying the number of operators in the shop has also been tested. In addition to this, effect of higher arrival rates on the performance of different assignment rules has been considered. It has been found that the performance of set-up - time - oriented rules (rules which give priority to the jobs which have minimum change - over time) improve with the increase of arrival rate.

CHAPTER 1

INTRODUCTION

Control over the detailed operations of large industrial firms have recently become a subject for basic research. An industrial system is a combination of many interdependent and interrelated sub-systems. A simplified model of the planning cycle of a manufacturing type of corporation is shown in Fig. 1. An important aspect of industrial control is deciding upon the precise use of manufacturing facilities at each instant of time. Several factors have to be taken into account in making these decisions, such as the availability of resources, costs of implementing the decisions and due dates. This type of decision making is called "scheduling". The present work considers only 'scheduling' sub-system of the planning cycle (Fig. 1).

Industrial scheduling problem differ greatly from one firm to another. Sometimes the manufacturing process consists of a series of operations at one work station on only one physical part; at some other times operations require very different labor skills and equipments on each of the many thousands of sub-assemblies. Sometimes inventories of finished goods are maintained to satisfy customer demands; at other times maintaining inventories becomes impossible under all conceivable circumstances. Unique

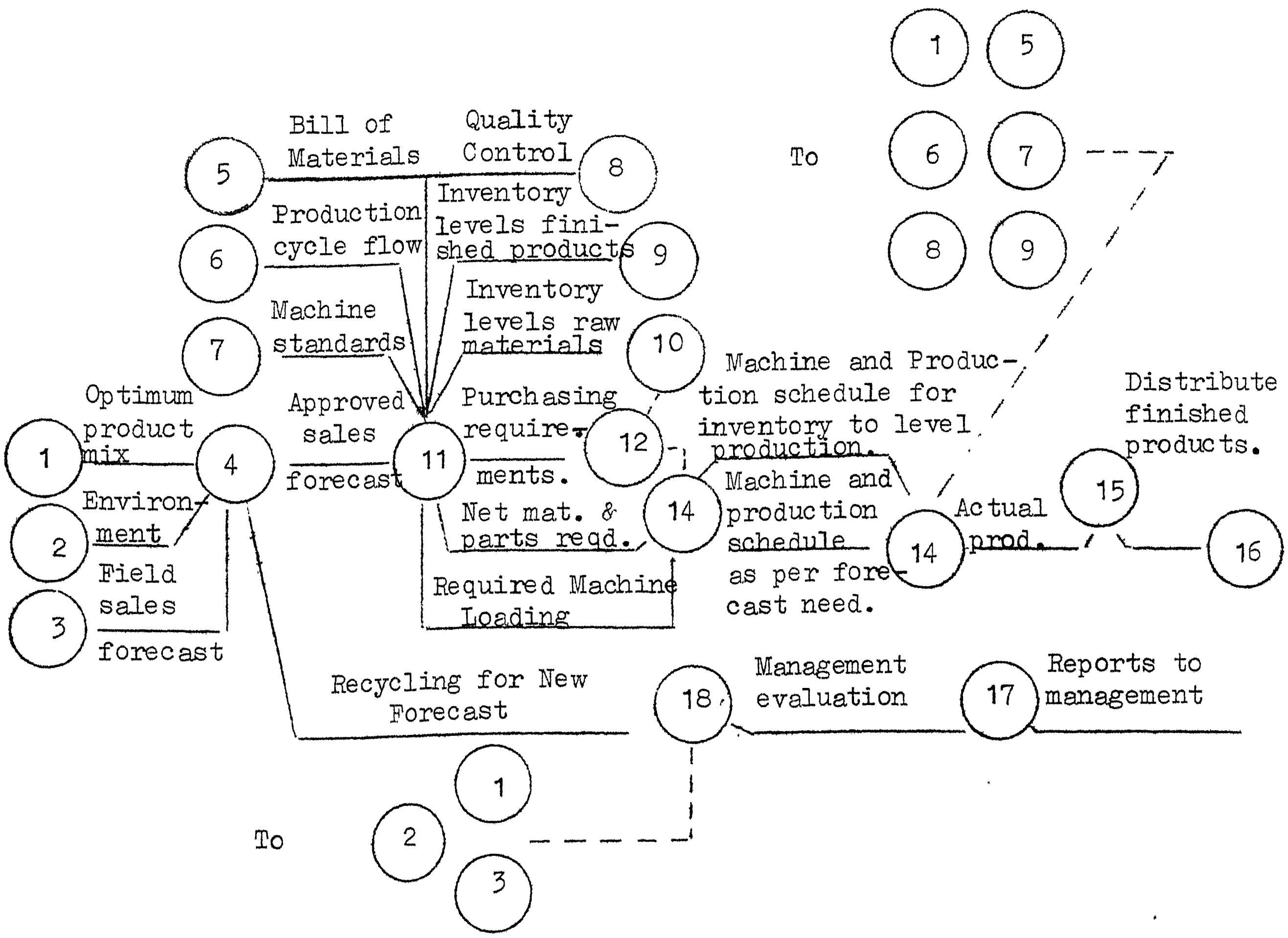


FIGURE NO. 1

CORPORATE PLANNING CYCLE
SIMPLIFIED MODEL

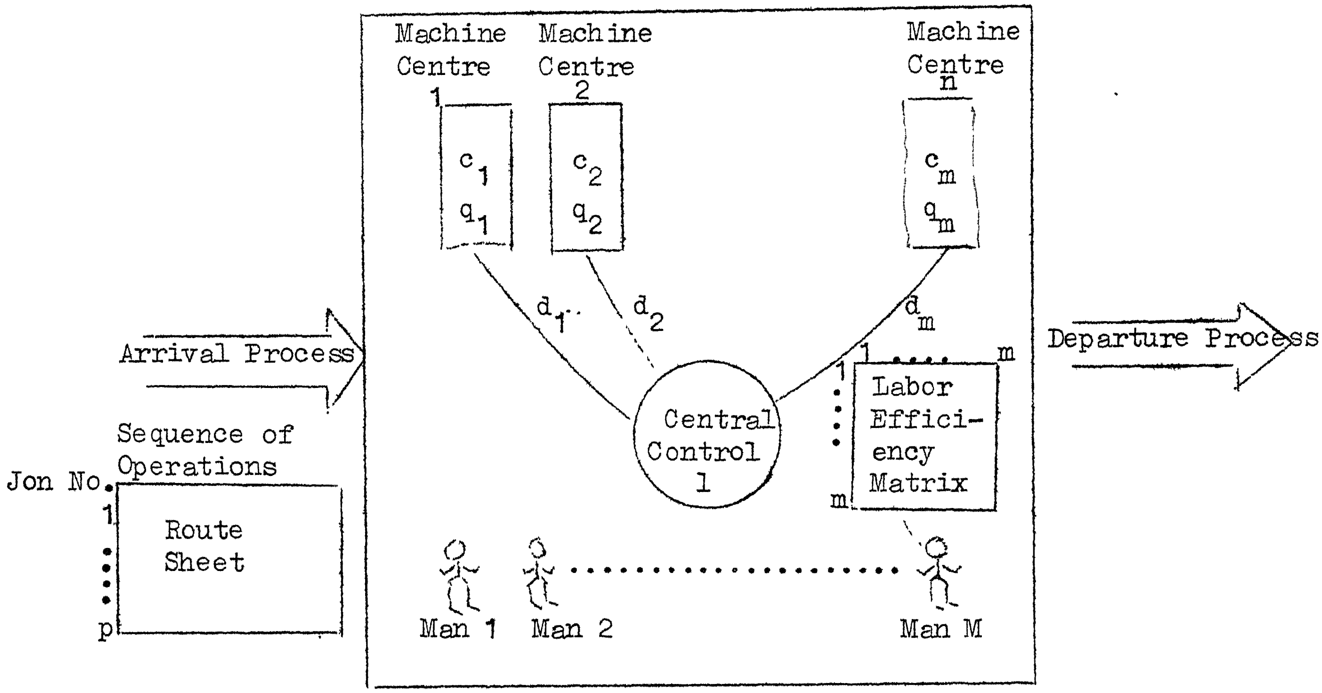
features of the firms organization, of the market, of plant capabilities are always present. Because of such diversity and complexity of industrial scheduling problems, it is impractical to account for every factor in any single analysis. A few special simplifications have so far been studied they are:

(a) Finished - Goods Inventory Control :

Individual items are stored until they are sold to the customer. The reorder quantity and reorder point is decided according to some criterion. Generally minimizing total cost (which includes inventory carrying cost, shortage costs and set - up costs) or maximizing the total profit (which is the difference of selling price and the total cost) or maximizing rate of investmet (which is profit / investment) are used as criteria to find out how much to produce and at what time of a particular product cycle.

(b) Project Scheduling :

A single project is to be undertaken, consisting of a large number of individual jobs. Every job has a set of immediate predecessors which must be completed before the job can be taken for processing. Thus there are some jobs which can be processed simultaneously while there are others which are to be done serially. The problem is to schedule individual jobs in such a way that the whole ...



c_i = Number of identical machines in the i^{th} machine centre. $(i = 1,\ldots m)$

m = Total number of machine centres in the system.

n = Total number of operators in the system.

n_{ji} = Efficiency of the j^{th} operator on the i^{th} machine $(i = 1, \ldots m, j = 1, \ldots n)$.

d_i = Degree of centralised labor assignment control exercised at the machine centre i. $(i = 1, \ldots m)$.

1 = Machine Centre selection procedure used in central control.

q_i = Queue discipline used at the i^{th} machine centre. $(i = 1, \ldots m)$.

= Mean arrival rate of jobs.

u = Mean departure rate of jobs.

FIG. 2

THE SCHEDULING SUB-SYSTEM

project is completed by the due - date if possible. This type of scheduling is also called "Network Scheduling".

(c) Job Shop Scheduling Problem :

A job shop consists of several work centres with different capabilities. If required, a work centre may be a collection of many identical facilities. Jobs arrive to the shop at random or with a specific probability distribution. A job has a fixed routing, and the due dates are fixed. The problem is to schedule the jobs to machines in order that

(i) due dates are met whenever possible or the job lateness is minimised,

(ii) the total time to complete all jobs is minimized, or

(iii) some other feasible criteria.

A schematic view of the job - shop scheduling sub-system is shown in Fig. 2.

The system consists of m machine centres and n labourers. Each machine centre is comprised of c_i identical machines. The jobs arrive at the system with a specified distribution. Each job is described by a job routing which specifies the machine centres required to completely process the job and the sequence in which they must be employed. The problem is to schedule the jobs and the operators in order to satisfy some criterion.

To decide a criterion for scheduling problems is difficult task. Scheduling problems are multiple - criteria problems and most of them are conflicting to each other. For example, one type of priority rule for job assignment may give less job lateness but it may increase the idle time in shop which ultimately results in higher in-process inventory costs. A system of weightages has to be decided to get an integrated objective function. The weightages depend on the type of firm and relative importance of due dates etc.

The job - shop studies have been focussed on machine limited system; systems whose performance is in no way limited by labor resources. Conway (4) and others have simulated only a machine limited system. Machine and labor limited system - where performance depend on both machine and labor resources, present new dimension to the problem with respect to the degree of control in work assignment. The freedom afforded by the queue disciplines (job assignment rules) is enhanced by control over labor assignment procedures. Labor assignment procedures take into account the quantity and quality of the labor force, which are relevant factors affecting system performance. The quantity and quality of labor are inter-related factors in the sense that various combinations of the two can result in equivalent performance. Increasing the size of work force reduces the degree of freedom afforded in work assignment procedures,

thus decreasing the relative effectiveness of a highly flexible work force. Thus considering labor and machine limited system, a wide range of practical design and control decisions such as the nature (skilled, semi-skilled or unskilled) and size of the labor force, the type and extent of training programs to develop labor flexibility, the degree of centralized labor assignment control, and the detailed assignment procedure etc., can be taken.

Whenever a job completes service on a machine, set-up time is required to make the machine ready to process next job. The set-up times will depend on the last job completed on the machine and the next job which is to be processed on the machine. Most of the workers have assumed the set-up time as sequence independent. In practical situation, the set-up times are highly random and the set-up times will be varying with sequence of jobs. In some situations the set-up times (i.e. set-up cost) may be so important that the criterion of minimizing set-up times may override other criteria such as minimizing job lateness, minimizing idle time etc. The present work compares the performance of set-up time rules of job assignment against the static and dynamic rules for job assignment which do not consider any significance of set-up times.

Thus the job-shop scheduling problem is to schedule jobs and operators to various machines in order to satisfy a chosen criterion. To choose a criterion is a difficult

in nature. Which of the criteria should be given more weightage, is a question which can be answered only if a firm and its environment are analyzed critically.

In this work following criteria are studied - (a) Minimizing the flow time per job, (b) minimizing job lateness, (c) minimizing idle time of a job, (d) minimizing job tardiness and (e) minimizing total jobs late. Machine utilisation and operator utilization are compared for different combinations of job assignment rules and labor assignment rules. Rules considered for assigning a job to a machine are :

Select a job having

(a) the shortest service time (SST),

(b) the nearest due - dates (DUDT),

(c) the least set - up time (LST),

(d) a set-up class which follows the set-up class of last job completed on the machine in optimal set-up sequence (optimal set-up sequence is found by solving the travelling salesman problem, it is discussed in Chapter 4). Ties are broken by selecting a job with nearest due date,

(e) same as (d) except that ties are broken by selecting a job with minimum service time,

(f) the shortest process time (process time is the sum of set-up time and the service time),

(g) the least slack/operation, and

(h) the largest service time.

Whenever an operator is free and there are many machines which need this operator, then the machine is chosen according to Labour Assignment Rules. Following two rules for Labour Assignment are studied;

(a) A machine which started demanding an operator earlier is given priority (FDFS), and

(b) A machine which has the maximum number of jobs waiting for service is, given priority (MAXQ).

The problem is discussed in detail in Chapter 3.

CHAPTER - 2

LITERATURE SURVEY

Scheduling problems involve so many variables that it is difficult or rather impossible to solve scheduling problems completely which arise in different industries. Problems differ very much from one firm to another firm, one department to another department in the firm and also from one place to another place. It is not possible to develop a single method which can be used to solve scheduling problems of all firms. Workers have tried several methods to solve this problem.

2.1 Methods Used To Find Optimal Solution :

In these methods, the problem is simplified by making some realistic assumptions, ignoring variables whose effect is negligible for all practical purposes. Some of the methods used are discussed below -

Johnson (9) developed a method which can give Optimal Solution for 2 machines N job problem. He assumed that the sequence of operation on all jobs is the same. He also extended his algorithm to solve a three machine N job problem, with some restrictions. In actual situations number of machines are large and also number of operations are different on different jobs. His algorithm has limited applications.

The other approach to the problem that is guaranteed to give an optimal solution is complete enumeration, which has been studied by B. Giffler and G.L. Thompson (25). Their first effort was to devise an algorithm for the complete enumeration of all "active" schedules. An active schedule is a feasible schedule with (a) no machine is idle for a length of time sufficient to process completely a commodity simultaneously idle, and (b) whenever an assignment of a commodity to a machine has been made its processing is started at the earliest time that both the machine and the commodity are free. All the active schedules are tested and one which gives optimal solution is selected. The solution by this method is computationally unfeasible. Giffer and Thomson solved a six job, six machine problem and the found that there are 84302 active schedules and computer time taken was 70 minutes. Real problems are impossible to solve on computer as they involve such large numbers of jobs and machines. Their method is applicable only for fixed technological sequence on all the jobs. They have developed a sampling technique which can be used to get approximate solution. Heller and Logemann (14) also developed an algorithm for construction and evaluation of feasible schedules.

Bowman (21), Wagner (23) and Manne (24) have tried to formulate the scheduling problem as an integer programming problem. These methods are also difficult to solve

computationally. Bowman estimates that formulating a simple problem involving three jobs and four machines in his terms would require an integer programming problem containing 300 to 600 variables and many more constraints. The formulation by Wagner of the problem would be of the same order of magnitude. Manne's formulation, the most compact of the three, would apparently require 51 variables and 94 constraints. In solving an integer programming problem by Gomory's cut algorithm, more constraints are added so that the ultimate number of constraints will be considerably larger. None of these authors claim that his formulation is computationally feasible (27).

McNaughton (12) developed an algorithm to minimize the loss associated by not meeting the deadlines. His algorithm is -

If the problem is such that the tasks can be scheduled without splitting in order of decreasing $r_i = p_i/a_i$, with no unused time before all tasks are finished, and with no task finishing before its deadline, then this schedule is minimal.

Where

P_i = penalty cost/time for i^{th} task

a_i = Processing time of i^{th} task.

His algorithm can be applied for one machine N job problem and it will give optimal schedule. But for M machine,

N job problem this algorithm becomes very complex. Even for a single machine, N job problem the conditions are rather restrictive and therefore it will not apply in many situations.

Schild & Fredman (15) modified McNaughton's algorithm to minimize linear loss function. Steps in the algorithm are as follows -

Step 1 : Arrange the tasks in order of increasing deadlines d_i. If in this arrangement all tasks are finished before their deadlines then the total loss is zero and problem is solved (situation B).

Step 2 : If situation B is not satisfied, arrange the task in order of decreasing r_i (r_i = p_i/a_i where p_i = penalty cost/time for i^{th} task, and a_i = processing time of i^{th} task). If in this arrangement every task finishes after their deadlines then optimal solution is achieved according to McNaughton's theorm (12). (Situation A)

Step 3 : If situation A does not exist, try to decrease the loss by using criterion D, applied to the first task j finishing before its deadline starting out with

$(j + 1)$. If for $(j + 1)^{th}$ task criterion D is not satisfied, try criterion D for $(j + 2)^{th}$ task and so on.

Criterion D :

$$\text{Let } \lambda_j = \left(\left| a_j - d_1 + a_1 \right| + (a_j - d_1 + a_1) \right) / \left(2 \left| d_j - d_1 + a_1 \right| \right) \tag{2.1}$$

$$\begin{aligned} j &= 0 \text{ when } a_j < d_1 - a_1 \\ &= \text{ when } a_j \geq d_1 - a_1 \end{aligned}$$

Then move task (j) ahead of task (i) if and only if,

$$p_1 (d_1 - a_1) \lambda j + P_1 \, a_j (1 - \lambda_j) - p_j \left| d_j - a_j \right| + (d_j - a_j) \sum_{i=1}^{j-1} (p_i \, a_j - p_j \, a_i) \tag{2.2}$$

where

p_i = penalty cost/time for i^{th} task

d_i = deadline for i^{th} task

and a_i = processing time of the i^{th} task.

If for $(j + 1)^{th}$ job, the criterion D is satisfied i.e. move job $(j + 1)$ in front of (j) and some tasks still finish before their deadlines repeat Step 3 till either all task finish after their dead lines, or all tasks except for a group of tasks at the end of the schedule finish after their deadlines.

Schild and Fredman have also tried to develop an algorithm for non - linear loss functions (15).

Gapp, Mankesar and Mitton (17) have developed an algorithm to minimize the in - process inventory. To minimize in-process inventory of the unconstrained problem, the jobs should be done in increasing value of c_i/p_i.

where

c_i = total cost associated with operation T_i

p_i = total time associated with operation T_i.

Thus the sequence of jobs to minimize in - process inventory is $c_1/p_1 \leq c_2/p_2 \leq c_3/p_3 \ldots \leq c_m/p_m$. Then they have developed an algorithm to take into account the precedence constraints.

2.2 Heuristic Methods :

To get an optimal solution for scheduling problems is computationally non-feasible for a size of problems which, arise in a general shop. Most of the time heuristic methods are used to get an approximate solution.

Performance of some dispatching rules is tested on a digital computer and one rule is decided which is simple to apply and easy so that operators may also understand it. The operator is given the instruction that he should select a job with shortest processing time or first come first

served or any other loading rule, out of the jobs waiting in queue for service of the machine. A complete description of loading rules is given in "Industrial Scheduling" by Muth and Thomson (27).

Another method is to develop procedures which need skilled operators. These type of rules are known as heuristic rules. One of the approach to develop heuristic rules is by levelling the production of the firm. The problem is to determine how best to respond to widely fluctuating customer demand. There are the following alternatives available to solve this problem -

(a) where manufacture for stock is possible, maintain the production rate and labor force constant. Customer demands are met by fluctuating inventories. In this case inventory will be large and may cause excessive carrying costs.

(b) Another approach is to change the production rate according to the demand. In this case the idle hours of operators will be very high, if labor force is maintained constant.

(c) The third approach may be to lay off or hire operators as required. This will increase the administrative cost and also there is no security of service of workers which may cause dissatisfaction in the workers.

Depending upon the relative weightages of different alternatives in a particular situations, one of three rules can be followed to meet fluctuating demand. In a similar way, houristic rules can be found out for different problems arising in scheduling. Which of the houristic rules will be better for a particular situation? Only experience can answer this question. The other difficulty with heuristic rules is that they may give a very bad result sometimes and hence care should be taken to use these rules.

Palmer (26) has developed a houristic method which will give near optimal solution for a problem which has fixed and same technological ordering for all the jobs in the shop. He defines a slope index for each job as -

$$S_p = -\frac{n-1}{2} A_{p1} - \frac{n-2}{2} A_{p2} - \ldots\ldots + \frac{n-3}{2} A_{p(n-1)} + \frac{n-1}{2} A_{pn} \qquad (2.3)$$

where

S_{pi} = Slope index for i^{th} job,

n = Number of machines (which is equal to number of operations also).

A_{pi} = Processing time for i^{th} operation (on i^{th} machine)

Assign the jobs in descending order of S_p's i.e.

$$S_{p1} \geqslant S_{p2} \geqslant S_{p3} \cdots \geqslant S_{pm}$$

where

m = number of jobs to be scheduled.

His criterion is to minimize the total elapsed time for m jobs. He has not considered re-routing (he has given a modification for his algorithm to take into account by-passing). Campbell, Dudek and Smith (22) developed another heuristic algorithm to solve N job M machine sequencing problem, where each job follows the same technological order of machines. By - passing re-routing is not considered. This method develops (N - 1) auxiliary two machine N job problems and then these problems are solved by using Johnson's two machine, N job algorithm. The solution which gives minimum elapsed time out of these (N-1) sequences is taken as the solution. Auxiliary problems are generated as follows -

for the k^{th} Auxiliary problem.

$$\Theta_{j1}\, k = \sum_{i=1}^{k} t_{j1} \tag{2.4}$$

$$\Theta_{j2}\, k = \sum_{i=N+1-k}^{N} t_{ji} \tag{2.5}$$

where

$\Theta_{j1}\, k$ = processing time for the j^{th} job on machine 1 of auxiliary 2 machine problem.

$\Theta_{j2}\, k$ = processing time for the j^{th} job on machine 2 of auxiliary problem.

t_{ji} = processing time of i^{th} operation on j^{th} job.

This method gives better results than Palmers but computation time is more.

2.3 Digital Simulation

All the methods used for getting an optimal solution or approximate solution are either impossible to solve computationally or have some limited application. For example, heuristic algorithms are applicable only for a limited class of problems where all the jobs have the same technological sequence and there is no re-routing and/or bypassing allowed (Palmer (26) has given a method to include by passing in a rudimentary manner, and that he has not considered any re-routing). Practical problems involve a lot of by passing and / or re-routing. No job-shop has all the jobs with same technological sequence. Another difficulty with these methods is that they provide optimal or near optimal solution for one criterion only, but scheduling is a multi - criteria problem. A system of weightages is decided for different criteria and then a combined objective function is defined. To resolve such complexities one has to take recourse to digital simulation.

The simulation of production shop requires the information about machines availability, labor availability sequence of operations and operation times on individual

jobs, the arrival time distribution of jobs to the shop, the due dates of jobs etc. The jobs are processed through the shop employing the priority rules under study. The arrival time, completion time and due dates of each job are noted down. Then, different priority rules are compared according to the chosen criterion. For example, if the criterion is to minimize penalty cost, then one has to minimize the penalty time which can be calculated as the difference of actual completion time of a job and its due date. Similarly in process inventory cost can be minimized by minimizing the difference between total time for which the job was in the shop (Time for which the job was in the shop is the difference between the completion time of a job and its arrival time to the shop) and the total processing time (total processing time for i^{th} job = $\sum_{j=1}^{N} t_{ij}$ where t_{ij} = operation time for j^{th} operation on i^{th} job, N = Number of operations on i^{th} job).

R.W. Conway (4) has simulated a shop which consisted of 9 machines which were continuously available without the designation of shift or days. The inter arrival time is exponentially distributed. Set up times were assumed to be independent of the sequence in which operations were performed and considered to be included in the given processing times. The service times are exponentially distributed. Conway tested the rules - Random, FCFS (First

come first served), FOPNR (has the fewest operations remaining to be performed), SPT (shortest processing times), LPT (longest processing times), MWKR (has the most work remaining to be performed), TWORK (has the greatest total work), in order to minimize in process inventory. He got the following results -

1. There does not appear to be any reasonable measure of performance in a job shop that is invariant under the choice of priority rules.

2. The rule SPT clearly dominates all the other rules tested. It's performance under every measure was very good. It is simpler and easier to implement than the rules that surpass it in performance.

3. There is no basis for believing that highly precise estimates of processing times are required for scheduling purposes. Both scheduling and wage incentive systems are often cited as the reasons for conducting the time study of operations.

4. It would be both worthwhile and possible for manufacturing firms to use a priority rule for job dispatching.

Conway (5) also simulated a shop taking due dates of jobs into account. He tested the performance of SPT, minimum due date, minimum slack, LPT, FCFS etc. and he found out that DUDT and SLACK gives very low variance estimates for the job lateness. Mean value of job lateness is minimum for the SPT rule. A combination of SPT and SLACK per operation turned out to be the best as far as tardiness is concerned.

Baker and Dzielinski (8), Simon (10), Gere (20) and a few others have also simulated a job shop. None of them has considered set-up time as sequence dependent.

Wilbrecht and Prescott (2) developed a method to take into account the set-up times alongwith process time. According to their simulation experiment, each job is given a set-up weightage randomly from a uniform distribution and the values are 1, 2, 3 or 4 units. The actual set-up time for a job is not computed until the job has been selected from a queue to be run on a machine. At that instant, the set up time is determined to be the absolute difference between the set-up value of the completed job and that of the new job. Thus, the set-up time for a job getting on the machine is made to depend upon the state of the machine which, in turn, depends upon the type of job which was previously on the machine. As an example, if the set-up value of the completed job was 1 and that of the new

job was 3, then actual set-up time for new job would be 3 - 1 = 2 time units. Other assumptions are similar to those of Conway's study. They tested the performance of Random, Due - date (earliest), Simset (jobs with similar set-ups have priority), shortest run time, shortest processing time against the criteria of minimum number of jobs waiting in queue, minimum in - process inventory etc.

Baker (3) simulated a shop with set-up time as sequence dependent. When a job arrives to the shop, it is given a set-up class. Jobs in a set-up class have same set - up times with another class of jobs. Thus, it is equivalent to say that similar type of jobs as far as their set-up requirements are considered, are grouped together. The measure of performance was the average time spent in the shop by completed jobs under steady state conditions. He tested the performance of FCFS (first come first served), SST (shortest service time), FIXSEQ (the sequence is fixed by following the sequence obtained from solving travelling salesman problem), MINSEQ (the job with the minimum set-up time with respect to the previous job is selected from the queue), a combination of SST and set-up time oriented rules, SPT (shortest process time which is equal to the sum of service time and set - up time). He found out that performance of SPT is the best as far as flow time is concerned. The set-up time oriented rules gives low variance, an indicator which reflects less tardiness. The

performance of set-up time oriented rules was better for secondary measures as maximum flow time. Also FIXSEQ and its extension allows the highest proportion of idle time.

All the above mentioned authors simulated a machine limited shop. They have assumed that labor force is unlimited which is not a realistic assumption. Rosser T. Nelson (1) analyzed a production system which is both machine limited and labor limited. He simulated a shop which has m machine centres and n operators. Each machine centre consists of c_i identical machines. Jobs arrive to the shop at Poisson's distribution. Jobs wait in a queue in front of each machine centre to be processed. When a machine is free, a job according to job priority rule is selected from the queue. Then an operator, if available is selected according to the labor assignment rule. The performance of combination of job assignment rule and labor assignment rule are tested for different number of operators. This method, thus decides also how much of labor force a company should have for a given arrival rate.

A similar digital simulation has been reported in this thesis.

CHAPTER 3

PROBLEM FORMULATION

3.1 System

The system to be simulated is shown in Fig. 2. Shop consists of M machine centres and N labourers. Each machine centre consists of Ci identical machines. The jobs arrive at the shop with a specified distribution. Information about number of operations, sequence of operations, due date, operation times on each job is known in advance.

3.2 Parameters of the System and their estimation

3.2.1 Arrival Process

The arrival of jobs to the shop is assumed to be randomly distributed according to Poisson's distribution. Empirical observations of job shops and related operations have revealed that the arrivals of units to the various service departments approximated very closely to the Poisson distribution (26, 30). But there is another rational basis for using the Poisson process as a model for a real system input. The arrivals to a service department originate from completed services in other departments. Each process at every department has some distribution of the service times,

the divergence of events from the scheduled times randomizes the input process, an intrinsic characteristic of the Poisson's Law.

3.2.2 Machines

The number of machine centres found in actual practice in a shop is quite large. But it has been reported that results with number of machine centres about 9 in a job shop will be applicable for large shops also (4, 5). Conway (4,5) has shown that increasing the number of machine centres from 9 to 27 had no significant effect on the relative performance of the decision rules. Based on this result it was decided to have only 9 machine centres in the present model. Some manufacturing processes are reported to have exponentially distributed processing times (30). As all the machines need not be identical, the service times of each are assumed to be exponentially distributed with different means (1.0, 1.2, 1.5, 0.9, 1.3, 1.0, 1.2, 1.1, 1.4). It is assumed that each machine centre has only one machine, but if a particular type of machine is in greater demand then the number of machines can be increased in that group.

3.2.3 Labor Force

The labor force is assumed to be flexible in this model. The number of operators varied between 6 and 9, taking different integral values in separate simulation runs.

The relative efficiency of each operator at every machine centre is an input variable to the model. The relative efficiency influences the service time required for an operation. In particular, service time required for the processing of the k^{th} job at the i^{th} machine centre using the j^{th} operator is equal to t_{ki}/n_{ji}, where t_{ki} is the time for the k^{th} job at i^{th} machine centre obtained from the service time density function and n_{ji} is the relative efficiency of the j^{th} operator at the i^{th} machine centre. The relative efficiency of the j^{th} operator at the i^{th} machine centre, $n_{ji} = 0$ means inability of j^{th} operator to perform work at i^{th} machine centre. Procedure to select the values of relative efficiency is discussed in section 3.2.8.

3.2.4 Number of Operations and Sequence of Operations for a Job.

A job is equally likely to find its first operation on any machine group. On the completion of that operation the job is equally likely to move to any other machine group or to leave the shop (two consecutive operations on the same machine are not permitted, but a job can return to a machine after an intervening operation). Being generated in this way, the number of operations per job is a random variable where expected value is equal to the number of machine groups - in this case 9 (4). To avoid memory over flow of computer, it is decided to restrict the maximum

number of operations required for a job to 20. The theoretical probability that number of operations required for a job will exceed 20 is 0.122.*

3.2.5 Operation Times

Once the number of operations and sequence of operations for a job are known, the processing times of each operation can be generated by an exponential distribution with a mean equal to the mean service time of the machine on which the particular operation is being performed

$$t_{ij} = - \alpha_k \cdot \mathrm{Ln}\,(r_n) \qquad (3.1)$$

where

t_{ij} = The processing time of the j^{th} operation on the i^{th} job.

k = Machine on which j^{th} operation of i^{th} job is being performed.

α_k = Mean service time of the k^{th} machine.

r_n = Uniformly distributed (0,1) random number

Ln = Natural logrithm.

---

* Since the number of operations are binomally distributed, the theoretical probability of number of operations exceeding 20 is $(1 - 0.1 \sum_{i=1}^{20} (0.9)^i) = 0.122$.

3.2.6 Due Dates

In actual practice, due dates are decided by company executives in consultation with the customers. The jobs are accepted for processing in the shop only if the management is satisfied that the jobs can be completed within the deadline period. Accordingly the decisions are made about carrying inventories, having an overtime or sub-contracting to other concerns. The situation is reverse here. The processing times for each job are known and due dates have to be fixed according to the processing times. Conway (5) has given a method to fix the due dates.

$$d_i = St_{i,1} + \text{Expected value of number of operations} \cdot \sum_{j=1}^{N_i} t_{ij} \qquad (3.2)$$

where

d_i - the due date of the i^{th} job

$St_{i,1}$ - time at which the i^{th} job is ready for its first operation.

t_{ij} - processing time for the j^{th} operation on the i^{th} job.

N_i - Total number of operations on the i^{th} job.

The expected number of operations are 9 here. In actual practice all the jobs are not equally important, some of the jobs which arrive to the shop may have a very high priority. Hence to be more realistic, it is decided

that the due dates be fixed by

$$d_i = \text{Arrival time of } i^{th} \text{ job} + r_n \text{ . Total processing time of the } i^{th} \text{ job.} \quad (3.2)$$

where

r_n is a random number.

In order to make due dates more critical, it was assumed that the random variable r_n is uniformly (2,5) distributed random number. The relative performance of the priority rules will not be affected by assuming any values for lower and upper limits for the random variable r_n.

3.2.7 Set - up Times

In a real job shop all the jobs are different from each other and their set - up times will also be different. Set-up time of a job depends on the preceeding job completed on the machine in question. If every job is assumed having different set - up time, the computer has to handle a (n x n) set - up time matrix, where n is the number of jobs being simulated. In the present work, 2,000 jobs are simulated and hence it is impossible to handle this matrix on IBM 7044 Computer. To avoid this difficulty, all jobs are devided into 10 set - up classes. Two parameters are to be decided for set - up time. First is the set - up class of the job. When a job arrives to the shop, a uniformly distributed (0, 1) random number is generated. If

this number falls between 0 and 0.1 then set - up class of the job is 1, if this number falls between 0.1 and 0.2 then set - up class of the job is 2 and so on.

The other question to be decided is the set - up time matrix. The elements of this matrix s_{ij} give the set - up time of an i^{th} set - up class job if the preceeding job is of the j^{th} set - up class. $s_{ij} = 0$ for $i = 1, 10$, i.e. diagonal elements of set - up time matrix are zero. The elements of this matrix are generated by a uniform distribution whose lower limit is 0 and the upper limit is 0.4. The service times for each machine is different but the average is approximately unity and as such the maximum set - up time is about 40% of the average processing time of an operation.

3.2.8 Labor Efficiency Matrix

In practice, the efficiency of each operator on nt machines is known and these elements can be input es of the system. In this work the elements of labor ncy matrix were generated by uniform (0, 1) random It was assumed that 20% of the total elements (81 ase)are zero i.e. a particular operator may not work on some particular machines.

The labor efficiency matrix, η_{ij} is generated as

$$X_1 = r_n \text{ ((0,1) uniformly distributed random number)}$$

$$
\begin{aligned}
X_1 \leq 0.2 \quad &\text{then} \quad \eta_{ij} = 0 \\
0.2 < X_1 < 0.5 \quad &\text{then} \quad \eta_{ij} = 0.5 + X_1 - 0.2 \\
X_1 \geq 0.5 \quad &\text{then} \quad \eta_{ij} = X_1
\end{aligned}
\tag{3.3}
$$

where

η_{ij} is the relative efficiency of the i^{th} operator on the j^{th} machine.

3.3 Assumptions

A lot which arrives to the shop at a time is considered as an entity and called a JOB. This job is not split further. No machine may process more than one operation at a time. Each operation, once started must be performed to completion (no pre-emptive priorities are taken into account). Each job, once started, must be performed to completion (no order cancellation).

The machines of the shop are not subject to any random failures that make them unavailable; i.e. they are always available at all times in all shifts. They do not spoil jobs in process or cause any attrition of lot size. Preventive maintenance is also not considered.

Time required to perform each operation is finite and fixed. In the actual case the operation times are

highly random, but Muth (27) has shown that the statistical property of job times is not of primary importance in developing optimal schedules. Hence it is assumed that the operation times are known in advance and they do not vary later. Transportation times between operations are excluded.

Job routing for each job is also known in advance and no alternative routings are permitted. Before start of any operation all preceeding operations should have been completed (i.e. no lap phasing is allowed).

Variation of arrival rates due to promotional activities and sub-contracting is not considered. Control of service times by multiple labor assignment or overtime is ignored. The company does not have any incentive schemes and hence the service times are assumed to be same for all operators.

The due dates are known and they are fixed. It is assumed that the shop works for all days in a year and 24 hours in a day.

The system is assumed to be time invariant. All the parameters of the system remain constant throughout the time studied. In a practical situation the efficiency of machines will decrease with their age. Worker efficiency is also a time variant parameter. In the beginning, when an operator is new he will be inefficient as he does not

have extensive training, after some time his efficiency increases with experience in the shop; however beyond his prime, his efficiency decreases. These effects are not considered.

The parameters do not change with the condition of shop i.e. this is a linear system. The arrival rate is not affected by the back-log in the shop. Service times of the machines do not vary with the queue length of the jobs in front of the machine.

Queue Discipline

It is assumed that there are 9 queues of jobs, one in front of each machine whenever a job is completed on any machine, another job is selected according to priority rule from the queue and if an operator is also available, then this job and the operator are assigned to the machine. There is another queue of idle machines(may be idle either due to non availability of operator or when no jobs are waiting for service), whenever an operator is free and there is atleast one job waiting on idle machines then according to some priority rule (as discussed in section 3.4.2) a machine is selected to assign the operator.

The penalty - cost and in-process inventory costs are assumed as linear function of time for which the job is late and is waiting for service on a machine respectively. All the jobs are equally important as far as penalty cost, and in-process inventory carrying cost are concerned.

3.4 Priority Rules

3.4.1 Priority Rules for Job Assignment

In actual practice, a large number of priority rules are used. Conway (4, 5) has tested about 90 priority rules but he has not considered the set - up time oriented rules. This work considers the relative performance of set - up time rules as compared to the static and dynamic rules which do not consider any importance of set - up times. Since, meeting due dates is an important criterion, hence emphasis is given to dynamic rules. A priority index (PI) is given to each job and job with minimum priority index is selected for operation. The rules considered are as follows:

(a) Static Rules

The priority index of a job remains the same throughout the simulation run in such type of rules. Following rules are tested in this category

i) Shortest Service Time : (SST) : A job which has got the shortest operation time on the machine is selected for processing first

$$PI_i = t_{ij} \tag{3.4}$$

where

PI_i = the priority index of the i^{th} job.

t_{ij} = the processing time for the j^{th} operation (for which the job was

in the queue on the machine) on i^{th} job.

The job with least PI_i is selected for operation on the machine.

ii) Largest Service Time (LST) : Jobs with larger work content are given priority

$$PI_i = -t_{ij} \tag{3.6}$$

where

t_{ij} = operation time for the j^{th} operation on the i^{th} job.

iii) Due Date (DUDT) : According to this criterion, a job which has the most critical date is selected for operation.

$$PI_i = d_i \tag{3.7}$$

where

d_i = Due date of the i^{th} job.

(b) Dynamic Rules :

The priority index of a job changes with the completion of each operation when such type of rules are followed. These rules take into account, the dynamic situation of the shop. Following rule is tested in this category:

Minimum Slack Per Remaining Operation (SLACK/OPERATION)

$$PI_i = \left((d_i - c) - \sum_{i=J_i}^{N_i} t_{ij} \right) / (N_i - J_i + 1) \tag{3.8}$$

where,

PI_i = Priority index of the i^{th} job.

d_i = Due date of the i^{th} job

c = Clock time

t_{ij} = Operation time for the j^{th} operation of the i^{th} job.

N_i = Total number of operations required to complete the i^{th} job.

J_i = The present operation number of the i^{th} job.

Other rules in this category are minimum slack, minimum job slack ratio, minimum processing time left, minimum number of remaining operations, maximum number of remaining operations, maximum total work content remaining and many more. All of these rules are not considered here.

(c) Set - Up Time Oriented Rules :

These rules take into account the set - up times of jobs. Under this category the following rules are tested;

i) Minimum Set up Time Sequence (MINSEQ) : When a job completes service, the next job selected for operation is one which has the least set - up time. If a job of the same set - up class as that of the preceding job

is available, that job is selected for operation; otherwise a set - up class which has got minimum changeover time is found and jobs of that class are given priority. Ties are broken by due - dates i.e. selecting a job with minimum due date.

ii) Optimal Set-Up Sequence and Ties by Due Date : (FIXSETSEQ, ties by DUDT) : The problem of finding an optimal set - up sequence which will minimize the total set - up time is equivalent to the problem of solving a travelling salesman problem. An optimal set-up sequence specifies the set - up class which must follow a given set - up class. When a job of i^{th} set-up class completes service, and if a job of the same set - up class is available that job is selected; otherwise a job of the next set - up class which follows the i^{th} set - up class in the optimal sequence, is selected. If no such job is available then a job with the least set - up time out of the available jobs is selected for operation. Ties in all the cases are broken by selecting a job which has got the nearest due - date.

iii) Optimal Set-Up Sequence and Ties by Minimum Service Time Rule (FIXSETSEQ and Ties by SST): Same method is followed as in (ii) above and ties are broken by shortest service time rule.

iv) Shortest Processing Time (SPT) : Processing time is the sum of service time and set - up time of a job

Processing Time for i^{th} job = Service time of the i^{th} job + set - up time for i^{th} job.

The job with the shortest processing time is given the highest priority.

3.4.2 Priority Rules for Labor Assignment :

When a job completes service on a machine, an operator is free. In a labor limited system, there may be many machines which require an operator. Which machine is to be given highest priority, is a question to be decided. The following rules are considered for selecting a machine for service :

(i) First Demanded First Served (FDFS) :

The machine, which became idle first is selected for service first. If an operator is available(who has an efficiency of more than 0.5) then a job is selected for this machine. If there is no operator out of the available

operators, has an efficiency of more than 0.5 on this machine then the next machine is selected which demanded an operator later as compared to the first machine but earlier than all the other remaining machines demanding an operator. This process is continued till, either there is no idle operator left or no machine is in demand of an operator or all the idle operators have got zero efficiency on all the machines which are demanding an operator.

ii) Maximum Queue Length (MAXQ)

According to this rule a machine which has maximum queue length in front of it, is selected for operation if there is an operator available who has an efficiency of more than 0.5 on this machine. Otherwise the second machine having maximum queue length is selected. This procedure is repeated till, either there is no idle machine left (an idle machine is one which is idle due to non-availability of an operator and there are some jobs waiting for service on this machine) or there is no idle operator left who has got an efficiency more than 0.5 on any of the idle machines left.

3.5 Criteria Used :

The relative performance of different assignment rules depends on the criterion chosen. A particular assignment rule may perform better for one particular criterion;

while it's performance may not be as good for some other criterion.

It is difficult or well-nigh impossible to have one single criterion for scheduling problems. An integrated criterion can be formulated which takes into account relative importance of different criteria. The following criteria are considered in this work -

(a) Primary Criteria :

(i) Minimizing Flow Time :

Flow - Time is defined as the time for which a job remains in the shop.

(ii) Minimising Job Lateness :

Job Lateness for a job = completion time of the job - due date of the job. (3.9)

(iii) Minimising idle time of jobs :

Idle time of a job = Flow time of the job - total processing time of the job. (3.10)

(iv) Minimising Job Tardiness :

Job - tardiness is defined as the extra time taken for completion of a job over and above the due date.

Job Tardiness = Completion time - due date

(if completion time is more than due date). (3.11)

= 0 otherwise.

(v) Minimising number of jobs late -

A job is said to be late if it is completed after its due date.

(b) Secondary Criteria :

In this category following criteria are considered:

(i) Minimising operator utilisation for particular operator conditions.

(ii) Minimising machine utilisation for a particular operator condition.

CHAPTER 4

TRAVELLING SALESMAN PROBLEM

The objective functions are numerous in scheduling problems, as discussed in section 3.5 of 3rd chapter. Another aspect of sequencing arises when each new job assigned to a specific machine has a significant set - up cost. The objective of minimizing set - up costs might override the other objectives. The set-up time matrix used in this work is shown in Table 1.

Elements of set - up time matrix t_{ij}, are the set-up times when job j follows job i. All the incoming jobs are devided into ten set-up classes and each job in the set-up class has the same set - up time with respect to another class jobs. The character of this particular matrix is analogous to the well known 'travelling salesman problem'. A salesman has to complete his route to various cities in minimun time (cost) without repeating any city. Similarly, an ordering must be found for a set of jobs, where $t_{ij} \geqslant 0$ $(i \neq j)$ is the set - up time needed to change facilities from the i^{th} to j^{th} job, and it is not required that $t_{ij} = t_{ji}$. Problem is to find a sequence with no internal cycles that minimizes the total set - up time.

The Assignment Method, as it is ordinarily used can produce internal cycles. An exact solution can be

found by using branch and bound method (30). Little, Murty, Sweeney and Karel (31) developed the branch and bound algorithm to provide solutions of optimal tours. The problem is solved by usual Assignment Method and the optimal assignment and lower bound is found out. If there are no internal cycles in this tour, then this is the optimal schedule and the total set - up time (cost) is equal to the lower bound. If there are internal cycles in the solution of optimal solution obtained by Assignment Method, then some other technique is to be applied. In branch and bound method, the approach is to discard the solutions which can not be optimal. To do this, further calculations are made by identifying each zero in the matrix with the sum of its minimum row and minimum column values, which is an opportunity cost measure of not making that particular zero assignment. Brahching is performed from the highest opportunity element, say $(k_1 - k_2)$. A logical partition of sequences can be, those sequences which have the assignment $(k_1 - k_2)$ and those which do not. The opportunity cost of each possibility increase as other zeros in the set - up time matrix are analyzed and hence lower bound increases. The possibility which has the least lower bound is analyzed first. This is continued till optimal schedule is arrived at. If there are some possibilities left which have lower bounds less than the lower bound for optimal schedule, they are partitioned further till the lower bound exceeds or equals

the lower bound corresponding to optimal schedule.

The method is discussed in "Systems Management of Operation" by Martin K. Starr[30], as follows.

Say that a problem S, fully reduced, is being considered for solution by the partition routine. One is concerned with the selection of a particular path, say from i to j, that will be the basis for creating two new problems:

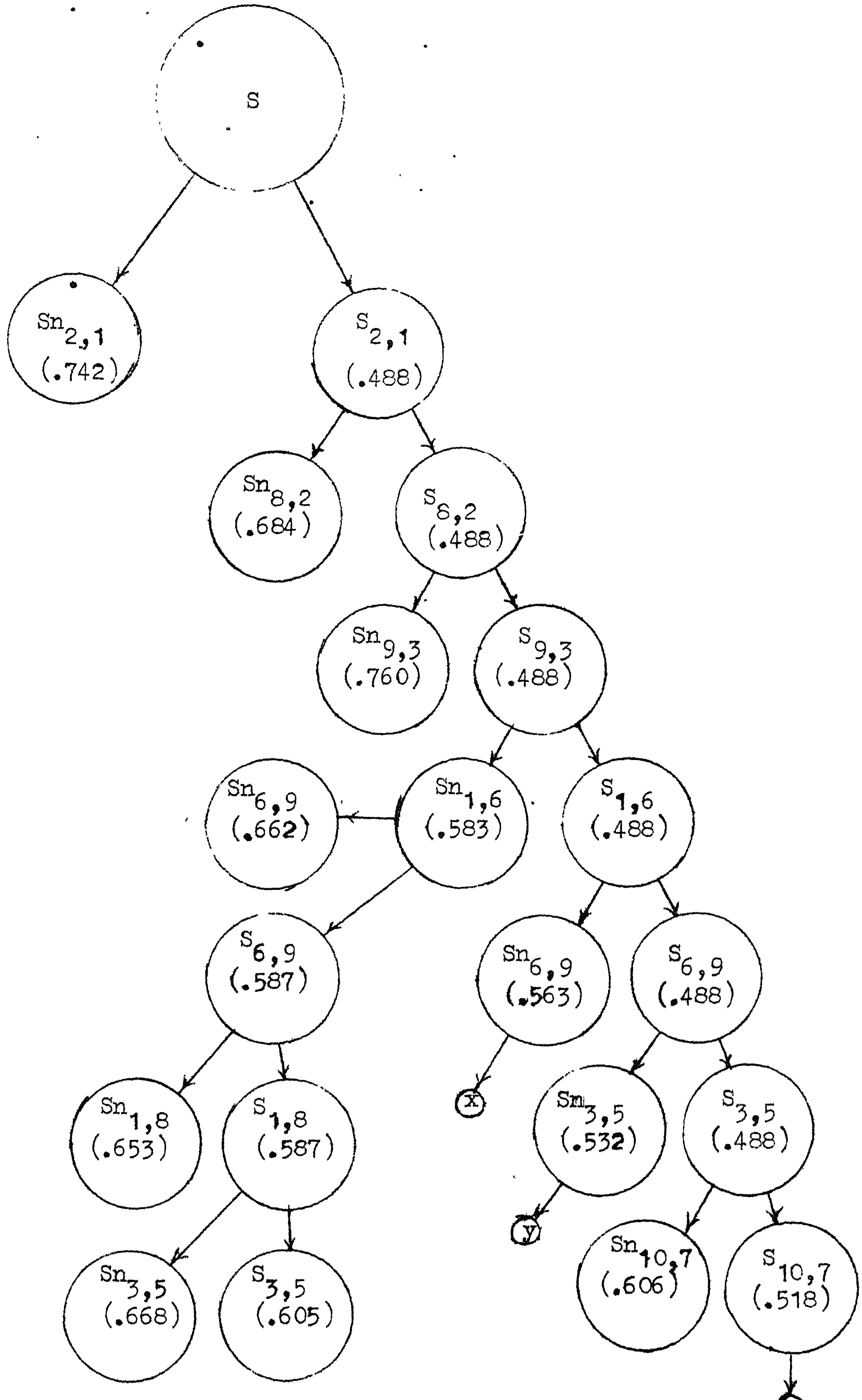
1. S_{ij}, the problem of finding the best solution from among all the solutions to S that include the step (i, j).

2. Sn_{ij}, the problem of selecting the best from among all the solutions to S that do not include the step (i,j).

Since in the problem S_{ij} it has been decided to go from i to j, one can prohibit going from i to any other city and prohibit arriving at j from any other city, by making all the entries in the i^{th} row and j^{th} column except s_{ij}, equal to infinity. One must also prohibit the future selection of the element s_{ji} making it infinite also, since a tour cannot include s_{ij} and s_{ji} and still visit all n cities before returning to the starting point. Since these prohibition may have eliminated some of the zeros of S, one can possibly further reduce S_{ij} and thus establish a new and greater lower bound on solutions of S obtained by solving S_{ij}.

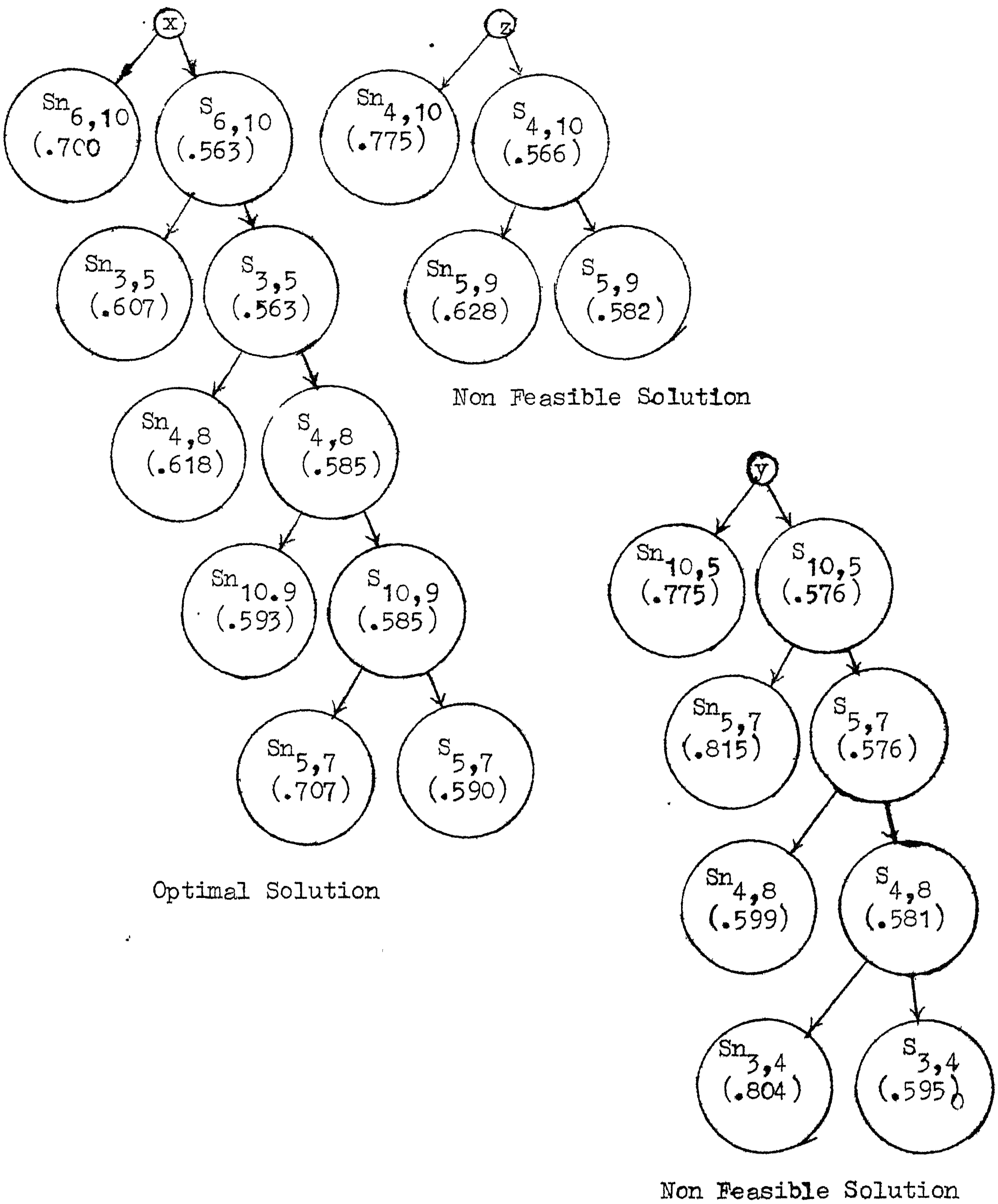
In problem Sn_{ij} one prohibits travel from i to j by making $s_{ij} = \infty$. Again a further reduction of this modified matrix may be possible that will give an increased lower bound on solutions obtained through problem Sn_{ij}. Now the objective of the selection of (i, j) is to make the lower bound on Sn_{ij} as great as possible in hopes that it can be discarded from the list of unsolved problems by the elemination routine without further partition. To accomplish this, one looks ahead at the reduction that will be possible in Sn_{ij} for each possible (i, j) and makes the selection such that the sum of two subsequent reducing constants will be a maximum. It should be obvious that it is necessary to consider only the zero elements of S as candidates, since if a non - zero element is selected no further reduction of Sn_{ij} will be possible.

Partition tree for the travelling salesman problem is shown in Fig. 3. A computer program for solving this problem is given in Appendix - 3. The optimal set-up sequence obtained is (1 - 6 - 10 - 9 - 3 - 5 - 7 - 4 - 8 - 2 - 1).



Non Feasible Solution

Fig. 3 (continued)



Optimal Sequence Is – 1–6–10–9–3–5–7–4–8–2–1

Fig. 3

PARTITION TREE FOR TRAVELLING SALESMAN PROBLEM

CHAPTER 5

COMPUTER PROGRAMME

FORTRAN IV language is used for programming the simulation experiment. A complete listing of the programme and flow charts are given in Appendix - 2. It is assumed that the maximum number of jobs in the shop are limited to 200 only. This restriction is placed because of limitations of computer memory locations available. First 200 jobs, which arrive to the shop are given numbers from 1 to 200. When 201^{st} job arrives, job which has completed service in the shop is found out. The 201^{st} job number is given the number of that job. This procedure is repeated for all the jobs. If none of the 200 jobs in the shop has completed its service before the next job arrives, then it is assumed that the arrival rate is very high and queue length practically becomes infinite.

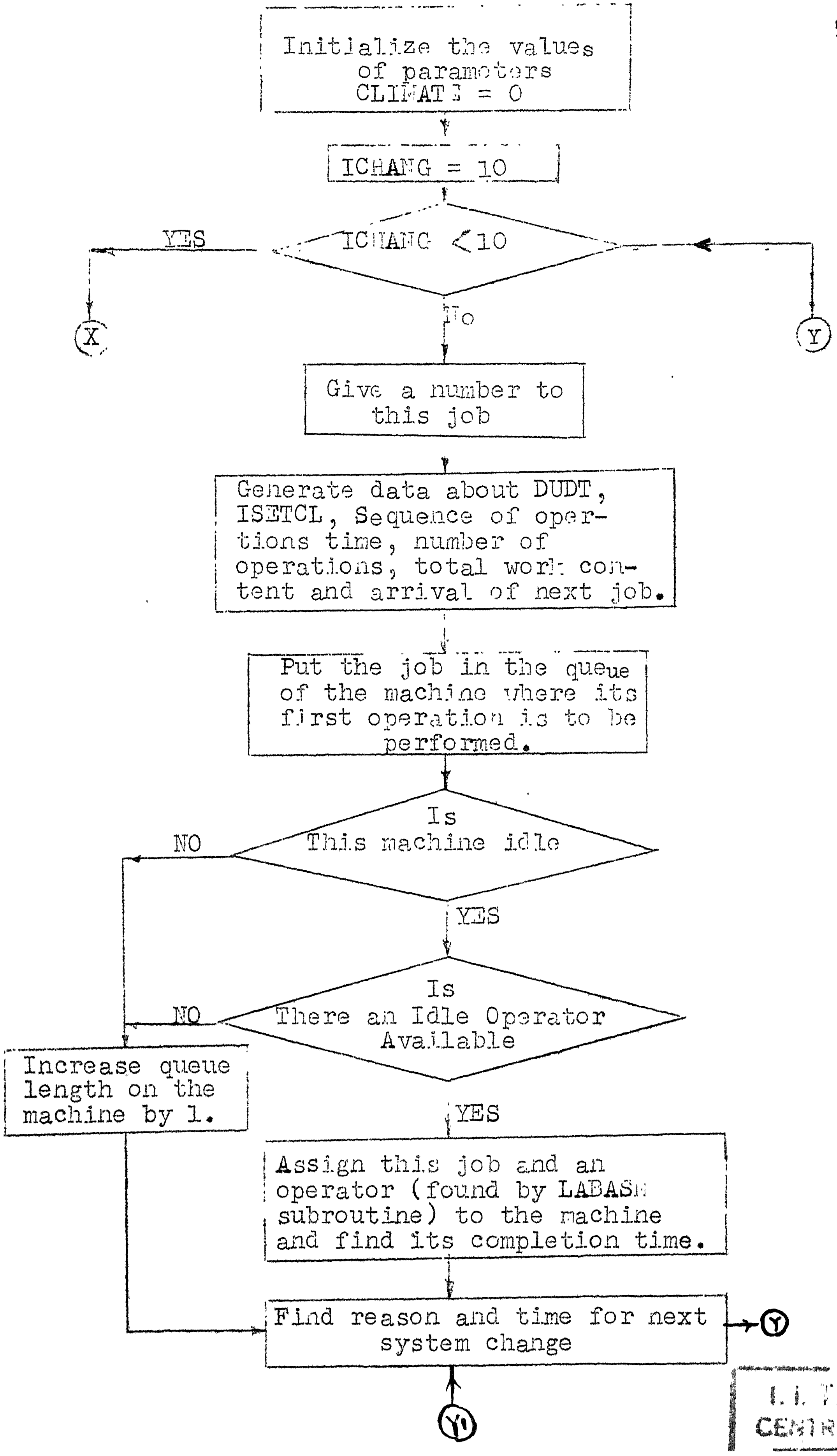
5.1 Main Programme :

The shop is assumed to be empty initially. First job arrives to the shop at zero clock time. When the job arrives, data about sequence of operations, operation times, due dates and set-up class are generated. The job is assigned to its first machine. The system will continue in this stage till a new job arrives to the shop or this job completes service on the machine.

The main programme is based on finding out the time and mode of next system change. The system may change in any one of the following modes -

(a) A new job arrives to the shop : Data for this job are generated by the DTGEN Subroutine. Then this job is assigned to its first machine, if the machine is free, otherwise it is placed in the queue of the machine.

(b) A job may complete its service on one machine: Whenever a job completes service, one of the operators is free and if there are some jobs waiting on the machine then a machine is available which demands an operator. If all the operations on the job have been completed, the results about completion time, penalty time, idle time, job tardiness are found out by ANLYS Subroutine. If all the operations on the job are not completed then the job is moved to the next machine. This is done by MOVJOB subroutine. The machine is selected for operation according to the labor assignment rule by the MCASN Subroutine. A new job is selected from the queue of the machine according to job assignment rule. An operator is selected and assigned to the machine. Again the time and mode of next system change is found out by SYSCHG Subroutine. Clock time is incremented by the time of next system change. This procedure is repeated for number of times till the shop completes


Initialize the values of parameters CLIMATE = 0
ICHANG = 10
ICHANG < 10
YES
No
X
Y
Give a number to this job
Generate data about DUDT, ISETCL, Sequence of opertions time, number of operations, total work content and arrival of next job.
Put the job in the queue of the machine where its first operation is to be performed.
Is This machine idle
NO
YES
Is There an Idle Operator Available
NO
YES
Increase queue length on the machine by 1.
Assign this job and an operator (found by LABASM subroutine) to the machine and find its completion time.
Find reason and time for next system change
Y
Y1

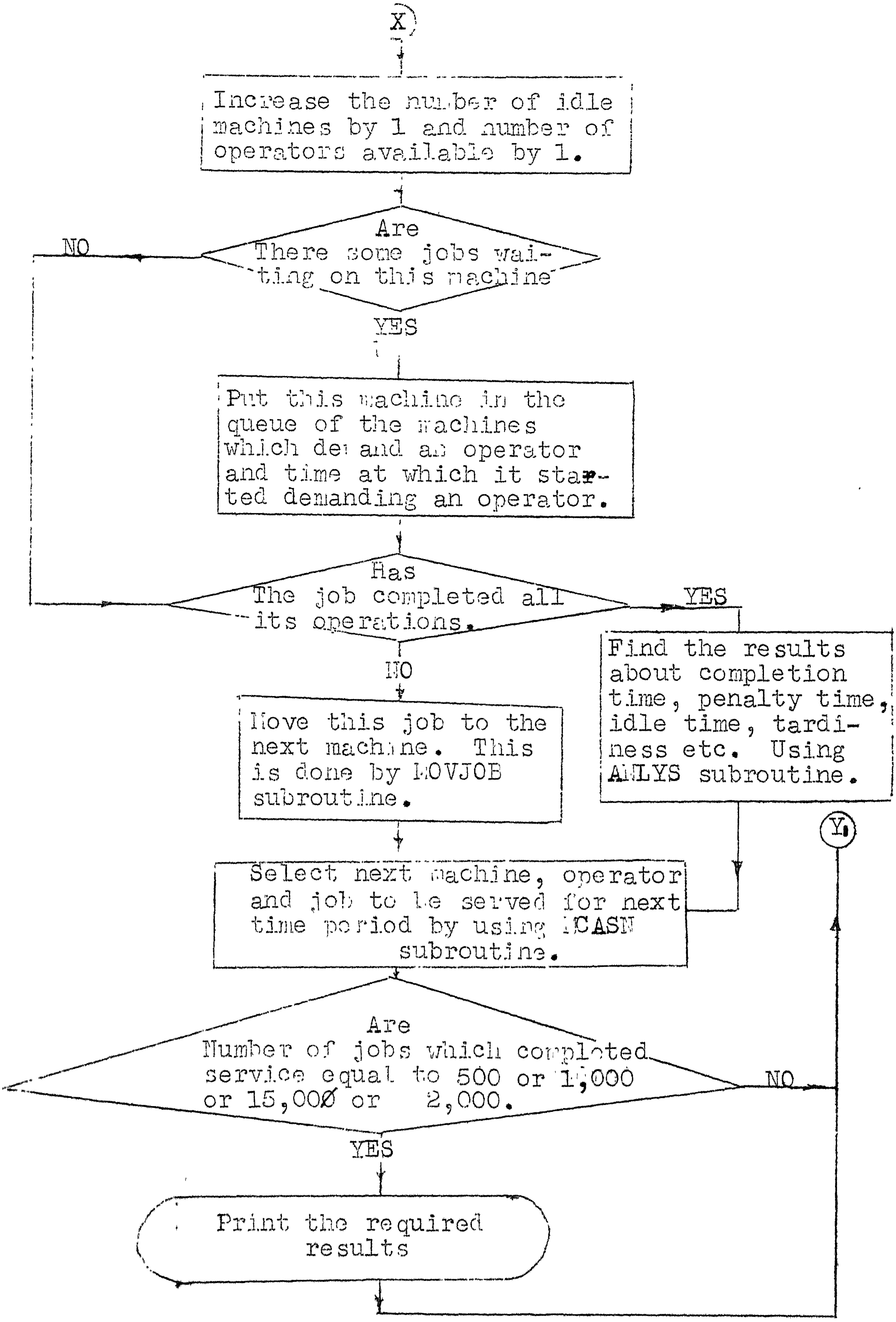


FIG. 4

MAIN PROGRAMME

1997 jobs. A flow chart for main programme is shown in Fig. 4.

5.2 Subroutines :

Several subroutines are used in this programme. A brief description of each subroutine is given below :

5.2.1 RANNUM [32]

This subroutine generates a random number from a uniform (0,1) distribution. Input to the subroutine is the "seed" and output from it are modified "seed" and a random number. A flow chart for the subroutines is shown in Fig.5.

5.2.2 DTGEN :

The procedure for generating values of parameters is discussed in Chapter 3. This subroutine generates data for sequence of operation, operation times, due dates and set-up class. Input to the subroutine are - job number (JOBNO) and seed (IX) for random number generation. Output from the subroutine are -

a. Job number of the next job which will arrive to the shop (IJK),

b. Arrival time of the next job number (TMARVL (IJK)

c. Sequence of operation of the present job (JOBSEQ (JOBNO,I), I = I, 20)

d. Operation times from an exponential distribution (TMJOB (JOBNO,I), I = 1,20)

e. Number of remaining operations which is equal to number of operations on the job (NRMOP(JOBNO))

f. Total processing time of the job, equal to summation of processing times, (TMTOT (JOBNO)).

g. Set-up class of the job, (ISETCL (JOBNO)).

h. Due date (DUDT (JOBNO)).

A flow chart of the subroutine is shown in Fig. 6 .

5.2.3 ASFMC

This subroutine puts the job in the queue of the machine on which its first operation is to be performed. Input to the subroutine are -

a. JOBNO - Job number of the job arrived.

b. NMC - Machine on which first operation is to be performed.

c. M1 - Maximum queue length at the machine. If there are M1 jobs waiting on the machine and a new job arrives whose first operation is on this machine, then it is assumed that queue length at this machine becomes infinite. The arrival rate is very high.

d. TMARVL (JOBNO) - Arrival time of job to the shop.

Output from the subroutine is -

a. JSTN (NMC, I) - An array which stores the job number of the job.

b. JOBOPR (NMC, I)- An array which stores the operation number of the job at machine NMC.

c. TMARIC (NMC,JOBNO) - Arrival time of job number (JOBNO) to machine NMC.

d. M1 - The value of I in JSTN (NMC,I) and JOBOPR (NMC I).

A flow chart of the subroutine is shown in Fig. 7.

5.2.4 SYSCHG :

This subroutine determines time and mode of the next system change. Inputs to the subroutine are -

a. IJOB - Job number to be given to the next job which will arrive to the shop.

b. TMARVL (IJOB) - Time of arrival of the IJOBth job.

c. N - Number of machines in the shop.

d. TMFNSH (I) - Time of completion of service on the I^{th} machine (I = 1,..N)

Outputs :

a. ICHANG - Index which shows the mode of next system change. If ICHANG = 10, then system changes due to arrival of a new job. If ICHANG = I, for I=1, 2, ...9 gives the machine number (I) where a job completes its service.

b. TMNXCG - Time at which next system change occurs.

A flow chart of the subroutine is shown in Fig. 8.

5.2.5 MOVJOB :

When a job completes service on a machine and it has not completed all of its operations, it is moved to machine on which next operation is to be performed. MOVJOB subroutine moves the job to the machine.

Input :

a. L1 - Number of operations completed on the job.

b. L2 - Operation for which it is moved to the machine.

c. K2 - Job number of the job being moved.

d. NOPTA - Number of available operators.

e. N1 - Max. queue length allowed at a machine.

f. JOBSEQ (NMC,I) Sequence of operations on this job (I = 1, 20)

g. CLTIME - Clock Time.

Output :

a. JSTN (NMC, I) - Array which stores the job number K2 on machine NMC.

b. JOBOPR (NMC, I)- Array which stores the operation number for which the job is moved to this machine.

c. TMARMC (NMC,K2)- Arrival time of job K2 at machine NMC.

d. NQUE (NMC) - Modified queue length of the machine.

e. IOL - Index which shows whether queue length exceeds the maximum allowable limit. (IOL = 0 if the queue length is less than N1, IOL = 1 otherwise).

A flow chart of the subroutine is shown in Fig. 9.

5.2. LABASH :

This subroutine assigns an operator to the machine which demands an operator.

Input :

a. NMC - Machine which demand an operator.

b. NL - Number of operators in the shop.

c. WEFFCY (I, NMC) - Operator efficiency matrix $I = 1, 2, \ldots NL$

d. IPNDX (I) - An array which shows operators which are already engaged in the shop. IPNDX (I) = -4 shows that the I^{th} operator is not free.

Output :

IOPER (NMC) - Operator selected for machine NMC. (IOPER (NMC) > NL shows that no operator is available which has efficiency more than 50% on this machine).

A flow chart of the subroutine is shown in Fig. 10.

5.2.7 JOBASN :

This selects a job from the queue of jobs waiting for service on a machine. Inputs to the subroutine are :

a. NMC - Machine for which the job is selected.

b. INDEXI - An index which gives the job-assignment rule, be used for selection of job.

c. N1 - The maximum queue length allowed at a machine.

d. Data concerning jobs such as processing times, due dates, set-up times, optimal set-up sequence, last job on the machine, clock time.

Output :

a. JOBI - The job selected according to a priority rule.

b. JL - Index used to find its position in the queue of jobs waiting on the machine NMC.

Flow chart of the subroutine is shown in Fig. 11.

5.2.8 MCASN :

This subroutine selects a machine out of the machines available which demand an operator. The machine is selected according to a Labor Assignment Rule. Input to the subroutine are :-

a. NL - Number of operators.

b. NOPTA - Number of available operators.

c. TMLSD (I) - Time at which various machines demand an operator (I = 1, N)

d. NQUE (I) - Queue lengths of various machines (I = 1, N).

e. INDEX 2 - The index which gives the Labour Assignment Rule to be used for selecting a machine.

f. Other data about jobs such as sequence of operations, processing times, due dates, set-up times etc.

Output is the selected machines and operators for processing machines. A flow chart for this subroutine is shown in Fig. 12.

5.2.9 ANLYS :

This subroutine is used to get results of the design variables. Input to this subroutine is the job which completes service in the shop and data about various parameters such as total processing times, due dates etc. Output from this subroutine are the values of various objective functions such as idle time, job - lateness, flow - time etc. for the job. The flow chart of the subroutine is shown in Fig. 13.

5.2.10 MIN :

This subroutine finds the minimum number in an array. Input to the subroutine is the array (one dimensional only) and number of elements in the array. The output is the number of the element which is minimum most in the array. A flow chart of the subroutine is shown in Fig. 14.

CHAPTER - 6

SIMULATION STUDIES

6.1 Experimental Conditions :

In a simulation experiment, the final results will largely depend on the initial conditions. If initial conditions do not represent the actual system, then the results will be unreliable and inaccurate. Thus, initial conditions should represent the conditions of the actual experiment.

This work compares the steady state performance of different Job and Labor Assignment rules in a job - shop. To study, the steady state behaviour, initial conditions can not be taken as zeros. Initial number of jobs in the shop should be equal to the estimated average number of jobs when steady state condition has reached. Difficulty is that the final number of jobs are not known in advance; they can only be found by performing the simulation experiment and then finding the average number of jobs in the shop in steady state. For next simulation run these many jobs are assumed to be in the shop initially. There is another technique to avoid this difficulty which is used to advantage in this work. Initially the shop is assumed to be empty i.e. no job is waiting for service in the shop and all the machines and operators in the shop are idle. The first job arrives to the shop at zero time. Then simulation experiment is

continued and statistics of various parameters are not noted down till 50 jobs complete service. A second experiment is performed and statistics of parameter are found after ignoring first 100 jobs. The two results are compared and if there is no significant difference, then it can be concluded that initially the shop is started with average number of jobs. It is to be noted that two different experiments are not necessary to find the statistics of the parameters of the objective function; in one experiment itself the statistics can be obtained by ignoring the first 50 jobs and then ignoring the first 100 jobs. Results for different criteria, using shortest service time rule for job assignment and FDFS for labor assignment are shown in Table 2. It is clear from Table 2 that the values obtained after ignoring first 100 jobs do not differ significantly from those obtained after ignoring first 50 jobs. For each simulation run, values of statistics for the above two conditions are tested to ensure that in the beginning, the remaining number of jobs in the shop is very nearly equal to the average value in steady state. If the two values are not same then another simulation experiment is performed and first 150 jobs are scheduled and their results are excluded from the final results. These results are compared with that obtained after ignoring first 100 jobs. If the two values are not significantly different, then it is again ensured as before

that initially the shop is fully loaded as defined earlier. In the present experiment ignoring first 100 jobs ensures that the shop is having average number of jobs in the beginning.

6.2 Number of Jobs to be Simulated:

Before deciding the number of jobs to be simulated, number of sequences of random numbers to be used must be decided. If transient behaviour of the system is studied, then a large number of sequences having length equal to the scheduled period must be used for simulation experiment. If steady state behaviour of the system is to be considered, then a large sequence (which ensures that the system has come to steady state) of numbers should be used (31). In the present work, steady state behaviour of the job shop is considered and hence one sequence of long length is used.

How many jobs should be simulated in index to study the steady state behaviour of the job shop? This is an economic problem which is decided by using sequential sampling techniques (31, 32, 33). If a smaller length of sequence is used then the system will not come to steady state and if a very long sequence is used, computer time will be unnecessarily wasted, which is very costly. In sequential sampling technique, first 500 jobs are simulated and results of design variables are found out, then

simulation is continued and when 1,000 jobs are completed, results of designed variables are again found out. If the results obtained in the two conditions do not differ significantly (a t-test will show whether they are statistically different or not), then the shop is assumed to be in steady state. If the results are statistically different, simulation experiment is continued further till 1,500 jobs complete service. Again the results obtained after simulating 1,000 jobs and that obtained after simulating 1,500 jobs are compared. If the two are statistically identical then the shop is in steady state and simulation experiment is terminated. If the results are statistically different then the simulation experiment is continued to complete 2,000 jobs in the shop. Again results of 1,500 jobs and 2,000 jobs are compared. If these two results are statistically different, then it is assumed that the shop will not come to steady state for all practical purposes. Then arrival rate is decreased (i.e. less jobs are made to arrive at the shop per unit time).

For all the rules tested, the shop comes to steady state within 2,000 jobs. Table 3show the results of different design variables after simulating 500, 1,000, 1,500 and 2,000 jobs. It is clear from the table that the shop comes to steady state after 1,500 jobs complete service.

6.3 Job Assignment Rules :

For the discussion to follow it is assumed that the mean inter-arrival time is 3 time units. When the number of operators in the shop are 9, there is a high probability that atleast one operator will be available on every machine. An operator may not be available only if the efficiency of the idle operator on the idle machine is zero. When the shop load is not high, there is very less probability that all the machines will be working simultaneously. Hence employment of any labor assignment rule will not affect the results significantly. Results for various Job Assignment Rules and Labor Assignment Rules for 9 operators are shown in Table 4. Mean flow - time for SST when Labor Assignment Rules is FDFS, is 20.7852 and when Labor Assignment Rule is MAXQ, it is 20.4577. Job lateness for DUDT rule when Labor Assignment Rule is FDFS - 10.9890; while job - lateness is -10.9142 when Labor Assignment Rules are approximately same for all objective functions tested.

Let us have a look at the Job Assignment Rules as to how each one behaves against various criteria.

6.3.1 Flow Time :

The results of this rule are tabulated in columns 1 and 2 of Table 4. SST and SPT are comparable and their performance is the best. SST and SPT give priority to the

jobs which can complete service faster i.e. they give priority to jobs which have less processing times. More number of jobs can complete service in the same time as compared to other rules and hence average flow - time will be less. The order of priority rules in increasing flow-time is SPT with an average flow - time of 20.6060 , SST with 20.7852 DUDT with 22.0168, MINSEQ with 22.2458, FIXSETSEQ with 22.6271 (Hence-forth FIXSETSEQ ties by DUDT will be referred to by FIXSETSEQ). The performance of FIXSETSEQ ties by SST is approximately same as FIXSETSEQ ties by DUDT), Slack/ operation with 22.6633 and LST with mean flow time 35.7753. Set-up time oriented rules will be better for high arrival rates i.e. when the shop is heavily loaded.When shop loads are moderate, there may not be enough jobs waiting in the queue having appreciably different set-up times. By selecting a job with minimum set-up time the saving in set-up times may be very small; on the other hand, a job with large processing time or a job having due-dates very far when selected, would deteriorate the performance of these rules. DUDT gives high priority to jobs having critical due-dates; this rule does not give any weightage to processing time. A job which has low work content but a far-away due date will have to remain in the shop for a long time. The flow-time will thus be more for DUDT and slack/operations.

6.2.2 Job Lateness :

Results for this criterion are tabulated in columns 3 and 4 of Table 4. Results show that the mean job - lateness is minimum for SST and SPT but their variance is high as compared to DUDT. For even 66% confidence level the values of job - lateness for DUDT lie below (-10.9890 + 1.[illegible] = - 10.9890 + 14.6621 = 3.6731) while for SPT the values of job lateness lie below (-12.3589 + 17.0368 = 4.6779). As confidence level increases the DUDT rule becomes more and more effective. Thus DUDT is the best rule for minimizing job - lateness out of the rules tested. The performance of slack/operation largely depends on the due - date fixation procedure. Conway (5) has shown that theoretically slack/operation should be better than DUDT but if different methods of due-date fixation are used, the performance of this rule varies. Sometimes this rule may work better than DUDT while on other occasions it may not perform as well.

Set-up time rules (MINSEQ and FIXSETSEQ) which do not give any weightage to processing times or due - dates of the jobs may be better for high shop loads. For EX=3 (i.e. average inter - arrival time 3 time units) set -up time rules (MINSEQ and FIXSETSEQ) are not better.

6.3.3. Idle Time :

Results for this criterion are tabulated in columns 5 and 6 of Table 4. Performance of SPT with mean idle time of 11.2677 time units and standard deviation of 11.353 and SST with mean idle time of 11.4467 and standard deviation of 11.6453 are comparable and their performance is better than the performance of the other rules. The variation of flow - time and idle time are identical. (discussed in Section 6.3.1)

6.3.4 Job Tardiness :

Results for job - tardiness are tabulated in columns 7 and 8 of Table 4. Performance of SPT and SST is better than the performance of other rules. Consider two jobs i and j with processing times a_i and a_j whose respective due - dates have already gone past. Let $a_i < a_j$; if the i^{th} job is scheduled before the j^{th} job, then the i job is late by a_i and the j^{th} job is late by $(a_i + a_j)$. Hence average tardiness is $= a_i + (a_i + a_j) / 2 = a_i + a_j/2$. If the j^{th} job is scheduled before the i^{th} job then average tardiness $= (a_i + a_j + a_j / 2 = a_i/2 + a_j)$. Since $a_i < a_j$, $(a_i + a_j/2)$ is in turn less than $(a_j + a_i/2)$. This result proves that if the jobs are done in order of increasing processing times then job tardiness will be less which means SST and SPT give better results compared to other rules.

6.3.5 Number of Jobs Late :

Results of number of jobs late for various priority rules are tabulated in the last column of Table 4. Performance of DUDT is the best with only 324 jobs being late out of 1997. DUDT gives the highest priority to the jobs which are critical as far as their due - dates are concerned and therefore there is less probability of a job being late. SST and SPT follow DUDT with 369 and 344 jobs completed respectively, later than their due-dates. The performance of slack per operation depends on the due - date fixation procedure. In this case, slack per operation follows the SST rule with 377 jobs completed after crossing their due dates. Performance of MINSEQ and FIXSETSEQ may be better if the arrival rate is high.

6.3.6 Secondary Criteria :

As discussed in section 3.5, the secondary criteria are the machine and labor utilisation. If machine and labor utilisations is lower corresponding to an Assignment rule then the performance of that rule is better. Same number of jobs have been completed by utilising machines and operators for a lesser time. For remaining time, the machines and operators may be utilised for some other purpose.

The set-up time rules (MINSEQ and FIXSETSEQ) are better compared to other rules for these criteria. Table 12

shows overall operator utilisation for different assignment rules. MINSEQ proves to be the best with an operator utilisation of 45.86 % followed by FIXSETSEQ ties by SST with an operator utilisation of 46.018%. To process all the jobs completely takes approximately the same processing time. The difference in operator utilisation among various rules is brought about by the difference in total set-up times. In set-up time rules the total set-up time is minimum and hence the performance of these rules must be better. Similar arguments hold for machine utilisation as well. Table 13 shows the variation of machine utilisation for different labor and job assignment rules. For NL = 9 i.e. when there are 9 operators in the shop, overall operator utilization and machine utilisation are same.

6.4 Labor Assignment Rules :

When an operator is free and there are many machines which need this operator, Labor Assignment Rules select a machine for that operator.

6.4.1 FDFS :

Whenever an operator is free, the operator will move to that machine which had demanded an operator first. This means that there is very less probability that an operator will continue work on the same machine for two consecutive operations. If there is no idle machine which

requires an operator except the one which has just completed its job, then the same operator will continue to work on this machine. This rule tends to distribute the operator's time on all the machines.

6.4.2 MAXQ :

A machine which has the maximum number of jobs in the queue is given priority. Consider an operator who has finished his service on the machine which has maximum queue length. Then according to FDFS, he has to move to a machine whose queue length is small. By the time the operator completes service on this machine all the jobs waiting in the queue of the former machine may become critical. And on the other hand, MAXQ tends to make the queue length of every machine approximately equal. A look at Tables 5,6 and 7 clearly shows that the performance of MAXQ is better than FDFS for all criteria.

6.5 Effects of Changing Number of Operators :

Number of operators are varied between 6 and 9. Average service rate of an operator does not change with number of operators in the shop. Hence, decreasing number of operators will increase the shop load. Performance of different labor assignment rules will be different when number of operators are less. Performance of MAXQ is better than that of FDFS (refer to Table 5, 6 , and 7) for

all criteria. In general, the performance of set-up time rules (MINSEQ and FIXSETSEQ) is relatively better than that of the other assignment rules as the number of operators decreases. This is because of the fact that the shop load increases with decrease in the number of operators. There is a long queue of jobs in front of each machine, whence, there is a high probability that overriding improvement can be accomplished by choosing a job with minimum set-up time rather than on some other criterion. The effect of varying number of operators on different objective functions is discussed in the following sections.

6.5.1 Flow - Time :

Mean flow time increases as number of operator is decreased. Standard deviation of the flow - time also increases with decrease in the number of operators. The mean flow time increases from 4 to 9% when the number of operators is reduced from 9 to 8 (Fig. 15 and Fig. 19). The level of performance of different job assignment rules remains the the same as in the case with 9 operators (Table 5).

When the number of operators is decreased from 8 to 7, there is an increase of about 11 - 15% in the mean flow - time. The level of performance of different job assignment rules also changes because of changes in the variance of set-up time (MINSEQ and FIXSETSEQ) rules.

Standard deviation of MINSEQ for NL = 7 is 24.7944 (Refer Table 6) while for DUDT it is 26.6660. For confidence level, MINSEQ rule gives better performance. The performance of MINSEQ goes on improving with increase in confidence level. Performance of SPT (with a mean flow time of 24.6757 and standard deviation of 22.7741) and SST (with a mean flow time of 24.7794 and standard deviation of 23.1080) continue to be better than that of all other job assignment rules (results for NL = 7 are tabulated in Table 6).

When the number of operators is decreased from 7 to 6, the mean flow - time increases abruptly (Fig. 15 and Fig. 16); there is an increase of about 28% - 30% in the mean flow time. The standard deviation of mean flow time also increases abruptly. The increase in standard deviation is as much as 35% for DUDT rule. This results on account of increase in shop load. The relative performance of MINSEQ and FIXSETSEQ continues to be better as compared to that of DUDT and slack per operation although SPT & SST continue to give better performance as compared to the set-up time oriented rules, MINSEQ and FIXSETSEQ (values of flow - time are tabulated in Table 7 for 6 operators in the shop).

6.5.2 Job Lateness :

Mean job - lateness increases as number of operators in the shop is decreased. On the other hand, the

standard deviation decreases when the number of operators is decreased from 9 to 7 but increases when the number of operators is further decreased (Fig. 16 and Fig. 20). The performance of DUDT for NL = 9 and NL = 8 are compared as follows

(i) For NL = 9 and 68% confidence level, we can say that the values of job lateness will lie below (-10.9890 + 14.6621 = 3.6731) (refer to Table 4). Against the same confidence level, values of job lateness will lie below (-9.2939 + 14.0342 = 4.7403) for NL = 8 (refer Table 5). These values show that for 68% confidence level NL = 9 gives lesser job lateness than NL = 8.

(ii) For 95% confidence level, the values of job-lateness for NL = 9 will lie below (-10.9890 + 2 x 14.6621 = 18.3352). Against the same confidence level, the values of job (lateness will lie below (-9.2939 + 2 x 14.0342 = -9.2939 + 28.0684 = 18.7745) for NL = 8. NL = 9 continues to have better behaviour.

(iii) For 99% confidence level; job lateness for most of the jobs will lie below (-10.9890 + 3 x 14.6621 = 32.9963) for NL = 9; while job lateness will lie below (-9.2939 + 42.102 = 32.8087) for NL = 8. This shows that NL = 8 gives smaller value of job lateness compared to NL = 9 with a confidence level of 99%.

For a higher confidence level 99.9% , NL = 8 will give less job lateness than NL = 9. Reducing number of operators, the performance of assignment rule improves with the criterion of minimizing job lateness. The mean job lateness increases by 7 to 17% when number of operators are decreased to 8 (Table 5). The order of performance of different job assignment rules continue to be the same as in the case of 9 operators.

When number of operators are decreased from 8 to 7 the mean job lateness increases by 21 - 30%; for different job assignment rules but the variance decreases still further. For 99% confidence the values of job lateness for NL = 7 will lie below (-6.5827 + 3 x 13.7043 = -6.5827 + 41.1129 = 34.5302), refer to Table No. 7. For NL = 8, the value of job lateness will lie below 32.8087 for the same confidence level, (these comparisons are done for DUDT). The job lateness for DUDT with NL = 8 is less than that of NL = 7, even for a high confidence level. The order of performance of different job assignment rules continues to be the same as in the case of NL = 9.

When number of operators are decreased to 6 , the mean job - lateness changes abruptly and it becomes even positive for most of the job assignment rules when FDFS is the labor - assignment rule. With MAXQ as labor assignment rule, the mean job lateness still continues to be

negative. The mean job lateness changes by 66 to 145%; when number of operators are decreased from 7 to 6. The standard deviation also increases as number of operators are reduced to 6. The order of performance of different job assignment rules still continues to be the same as in the case of 9 operators (Table 7).

6.5.3 Idle Time :

Mean idle - time and standard deviation of idle time increases as the number of operators are decreased in the shop (Fig. 17 and Fig. 21). When number of operators are reduced from 9 to 8, there is an increase of about 7 - 12% in mean idle time for different job assignment rules. Correspondingly there is an increase of about 14 - 16% in the standard deviation. The order of performance of different job assignment rules remains the same as in the case of NL = 9 (Table 5). The performance of SPT and SST are better than that of others. MINSEQ and FIXSETSEQ are next in the list. Performance of DUDT and slack per operation is also comparable to that of MINSEQ & FIXSETSEQ (these conclusions can be drawn from Table 5).

When number of operators are decreased from 8 to 7, there is an increase of about 19 - 24% in mean idle time for different job assignment rules (Fig. 17 and 21). The order of performance of job assignment rules still continues to be same as that in the case of NL = 8 (Table 6).

When number of operators are decreased from 7 to 6 there is an increase of about 30 - 42% in the mean idle time for different job assignment rules (Fig. 17 and Fig. 21). The performance of set - up time oriented rules is better as compared to DUDT and slack per operation. With NL = 7, the mean idle time of DUDT is less than that of MINSEQ but the standard deviation of MINSEQ is less than that of DUDT (Table 6). With NL = 6 , both mean and the standard deviation of idle time are less for MINSEQ (Table 7).

6.5.4 Job Tardiness :

Mean and standard deviation of job tardiness both increase with decrease in number of operators (Fig. 18 and Fig. 22). When number of operators are decreased from 9 to 8, the mean job tardiness increases by 8% for SST and SPT but for DUDT it increases by as much as 26%, when the labor assignment rule is FDFS. For MAXQ, the mean job tardiness increases by 4% for SPT; whereas it increases by 23% for slack per operation (Table 4 and Table 5). When the number of operators are decreased from 8 to 7, the mean job tardiness increases by 19 - 33% for different job assignment rules. The increase in the job tardiness for SPT and SST is the highest in this case (Table 5 and Table 6). The order of performance of different job assignment rule remains the same as in the case of 9 operators (Table 6). Performance of MINSEQ is comparable to that of SPT and SST.

When number of operators are decreased from 7 to 6 there is an increase in the mean job tardiness by about 24 - 52% for different job assignment rules. The variance obtained with MINSEQ and FIXSETSEQ rules turns out to be smaller than the variance obtained with SPT and SST rules With confidence level of 2σ; FIXSETSEQ gives lower job tardiness as compared to SST. As confidence level is increased the performance of FIXSETSEQ improves further with respect to the SST rule. Thus the relative order of performance of various assignment rules changes when number of operators is changed from 7 to 6. (Table 7). Performance of FIXSETSEQ is better than SPT and SST and performance of SST, SPT and MINSEQ are comparable in the case when number of operators is six.

6.5.5 Number of Jobs Late :

Number of jobs which are late increases as the number of operators are reduced. (Fig. 23 and Fig. 24). When number of operators is decreased from 9 to 8 the relative order of performance of different assignment rules does not change. The performance of DUDT with respect to minimizing number of jobs; deteriorates as the number of operators is decreased. For NL = 7, and labor assignment MAXQ, the SST and SPT rules perform better (Table 6). When the number of operators is 6, the performance of DUDT is not as good as the performance of MINSEQ and FIXSETSEQ (Table 7). SST and SPT continue to give good results for NL = 6.

Secondary Criteria :

It is only for the secondary criterion that performance improves when number of operators are decreased. When there are 9 operators in the shop the overall operator utilisation is approximately 46% for all the assignment rules (Table 12). If jobs continue to arrive at the shop with the same mean inter-arrival time then for more than 50% of the time, the operators would be idle. No firm can afford to pay an idle worker. Hence operator utilisation with varying number of operators is studied. Table 12 shows that operator utilisation is approximately 46% for 9 operators in the shop, 53% for 8 operators, 64% for 7 operators and 78% for 6 operators (plotted in Fig. 23 and Fig. 24).

For a particular operator condition, a rule which gives a smaller value for operator utilisation is better; as the same number of jobs have been completed by allowing an operator more of free time. In case of minimising operator utilisation for a particular operator condition the set - up time rules (MINSEQ & FIXSETQ) give the best results. With NL = 6, with the same number of jobs completed the operator utilisation is 76.836% in the case of MINSEQ whereas it is 79.42% for SPT (Table 12). Obviously, MINSEQ is better than SPT.

If all the operators have the same efficiency on each machine, the overall machine utilisation should be same

with different operator conditions. In this work, each operator has varying efficiencies on different machines. It may be possible that an operator may have more than 90% efficiency on one particular machine where others may have less than 70% efficiency. If this operator is turned out of the factory, then average time needed to complete service on this machine will increase. Thus with the decrease in the number of operators, the over-all machine utilisation will increase. Table 13 shows the variation of machine utilisation with different operator conditions for various combinations of labor and job assignment rules. The machine utilisation varies approximately from 46 to 52% for different job assignment rules when numbers of operators are changed from 9 to 6.

In general MINSEQ and FIXSETSEQ gives less machine utilisation as compared to other rules for the same operator condition.

6.6 Decision about Number of Operators :

As discussed in section 6.5, reducing number of operators in general increases the idle - time of jobs increasing in-process inventory costs and job tardiness (which causes penalty cost). When there are more operators, the idle time of operators is more. Idle time of operators is quite costly. Thus, deciding number of operators is an

economic question. One cost parameter increases (in process inventory, penalty cost), whereas the other parameter is reduced as the number of operators are decreased (cost of idle workers). The number of operators can be decided if cost of carrying in - process inventory, job lateness and idle labor is known.

6.7 Decision about which of the Operator is to be fired:

Depending on the situation let us assume that the management has taken aneconomic decision to reduce the number of operators by one. Now which of the operators should be turned out? This question is answered by knowing the individual operator utilisation. Individual operator utilisation with different operator conditions are tabulated in table 8, 9, 10 and 11.

As shown in table 8, operator number 7 has the minimum utilisation of only 7%. The operators are selected according to the maximum efficiency rule (operator which has maximum efficiency out of the idle operators is selected for operation). Operator number 7 has a small value of efficiency for most of the machines. It must be checked now if there is some machine which can be handled only by this operator i.e. the machine is a special machine and only one person is available to operate this machine. In such a situation, operator with the least individual utilisation can not be fired. In the present case operator 7 is not

a special operator and hence his services may be terminated.

Similarly, for different operator conditions, decision about laying off can be taken by critically analysing the labour efficiency matrix and individual operator utilisation.

6.8 Effect of Higher Arrival Rates :

As the arrival rate is increased, the performance of set - up rules (MINSEQ and FIXSETSEQ) improves as compared to the other rules. Since minimizing the job lateness is one of the most important criterion, hence only DUDT and MINSEQ rules are compared for a mean inter-arrival time of 2.00 time units (TX = 2.00). The results for different criteria are tabulated in table 14.

6.8.1 Performance of Assignment Rules Using Different Criteria :

Mean and standard deviation of flow time is less in the case of MINSEQ. The average job lateness of MINSEQ is less but the standard deviation is higher. Even for 60% confidence level; job lateness for MINSEQ is larger than that of DUDT. For minimizing idle time and job tardiness the performance of MINSEQ is better (only the standard deviation of job tardiness of MINSEQ, which is 44.0727, is higher than that of DUDT, which is 43.7481 as shown in

Table 14). But even for 39% confidence level; job tardiness for MINSEQ will be lesser than that of DUDT. DUDT gives minimum value in case of EX = 3.00 whereas for EX = 2.00, MINSEQ gives the smaller value for the number of jobs late (Table 14).

For secondary criterion of minimising operator utilisation and machine utilisation, performance of MINSEQ is better than DUDT (Table 15).

6.8.2 Effects of Changing the Number of Operators :

When number of operators are decreased the performance of MINSEQ is improved as compared to DUDT. In case of DUDT, it was not possible to have 7 operators because the queue length at machine number 3 increased to more than 100.

When the number of operators are decreased from 9 to 8, there is an increase of 40% in the mean flow time for MINSEQ and 65% for DUDT (refer to Table 14). The mean job lateness is increased by 271% for MINSEQ and by 372% for the DUDT Rule. Mean idle time is increased by 64% for MINSEQ and by 112% for DUDT. Mean job tardiness is increased by 47% for MINSEQ and by 69.5% for DUDT. Number of jobs late is increased by 2% for MINSEQ and by 26.8% for DUDT.

When the number of operators are reduced from 8 to 7, DUDT could not simulate 1997 jobs as the queue length

at machine 3 increased to more than 100.

The operator and machine utilisation are plotted in Fig. 27. The individual operator utilisations for different operator conditions are tabulated in Table 16.

CHAPTER 7

RESULTS AND CONCLUSIONS

7.1 Results:

A model is built to simulate a job - shop. Jobs arrive at the shop continuously. Each machine has a different mean service time. Processing times are generated from an exponential distribution. Due dates are fixed to take into account the processing times and also the urgency of a job. Jobs to be serviced on a machine are selected by job assignment rule, out of the jobs waiting for service on the machine. A machine is selected by employing a Labor Assignment Rule. From the simulation studies, following results are drawn :

1. The performance of SPT and SST is better compared to other rules for minimizing flow - time, minimising idle time and minimising job - tardiness when mean inter-arrival time of jobs to the shop is 3 time units.

2. Performance of DUDT is better for minimizing job lateness and minimising number of jobs late. The relative performance of DUDT deteriorates as number of operators are decreased in the shop.

3. In general, decreasing number of operators increases mean value of flow-time, job lateness, idle time, job tardiness and number of jobs late. The variance of different performance indices increase as number of operators are decreased (job lateness is an exception, where the variance decreases as number of operators is decreased). The variations in the design variables is not much when number of operators is decreased from 9 to 8. The variation becomes more when number of operators is decreased from 8 to 7. The values of different objective function changes abruptly when number of operators is decreased from 7 to 6.

4. The performance of Maximum Queue Length as a Labor Assignment Rule is better than First Demand First Served for all operator conditions and for all objective functions.

5. Operator utilisation increases as number of operators is decreased. Machine utilisation also increases as number of operators are decreased.

Performance of set - up time oriented rules (MINSEQ and FIXSETSEQ) is better against the

criterion of minimizing operator utilisation for same number of jobs completed within the same time.

7. Decision about which of the operators is to be fired when the management has decided to decrease the number of operators, can be taken by having a look at the labor efficiency matrix and individual operator utilisation. An operator with the least utilisation is laid off first.

8. The performance of set - up time oriented rules is better for higher arrival rates.

9. Although performance of SPT is better as compared to SST, it is difficult to apply the same in practice. SST is a static rule and operators can easily understand it. SPT is a dynamic rule, where each time the operator has to find the sum of processing times and the set - up times. Now the job with minimum process times is selected for processing. Thus, use of SPT entails the requirement of a skilled operator.

10. The set - up time rules are difficult to apply in practice because of their complexity.

7.2 Scope For Further Work :

The results reveal that a single assignment rule is not suited for all the criteria under study. The assignment rules which give priority to the jobs with critical due - dates are well-suited for minimising job - lateness; whereas, SST & SPT are best suited for minimising idle time. A combinatorial approach to job assignment rules can be made, which can take into account the critical due-dates as well as the processing time requirements for a job. The rules for scheduling thus formulated should be simple and easy to comprehend for the workers.

In the present work, the different criteria tested are independent of each other. In practice, however, an optimum satisfaction of all the criteria is to be looked for. The management is interested not in minimising penalty for a single criterion, but in over-all optimisation. An integrated objective function can well be formulated by considering the relative importance of various criteria (e.g. in-process inventory, job late-ness etc.) and the optimisation can be done subsequently.

A simulation approach as used here can well be used to advantage for the combinatorial problem.

APPENDIX - 1

T A B L E S

TABLE NO. 1

SET - UP TIME MATRIX

Set up class of preceding jobs	Set up class of following job									
	1	2	3	4	5	6	7	8	9	10
1	0.00	0.3[illegible]	0.31	0.20	0.31	0.13	0.31	0.13	0.28	0.20
2	0.01	0.00	0.13	0.15	0.17	0.12	0.14	0.29	0.36	0.30
3	0.35	0.23	0.00	0.12	0.04	0.29	0.33	0.09	0.19	0.31
4	0.31	0.27	0.39	0.00	0.17	0.39	0.31	0.02	0.20	0.04
5	0.36	0.35	0.20	0.06	0.00	0.23	0.05	0.06	0.21	0.28
6	0.37	0.35	0.16	0.21	0.37	0.00	0.23	0.13	0.08	0.00
7	0.25	0.28	0.38	0.04	0.35	0.12	0.00	0.09	0.18	0.03
8	0.33	0.03	0.38	0.37	0.19	0.27	0.38	0.00	0.24	0.22
9	0.16	0.06	0.02	0.15	0.29	0.35	0.23	0.19	0.00	0.36
10	0.24	0.15	0.35	0.29	0.07	0.23	0.10	0.19	0.24	0.00

TABLE No. 2

TESTING FOR INITIAL CONDITIONS
JOB ASSIGNMENT RULE - SST
LABOR ASSIGNMENT RULE - MAXQ

L = 9

EX = 3.00

Objective Function	Results Obtained after excluding first 50 jobs	Results obtained after excluding first 100 jobs
Minimizing Mean Flow Time	20.4804	20.4577
Minimizing Average Idle Time	11.1223	11.1210
Minimizing Average Job Lateness	-12.5706	-12.4966

TABLE NO. 3

NUMBER OF JOBS SIMULATED

JOB ASSIGNMENT RULE - SST

L = 9

EX = 3.00

No. of Jobs Simulated	Objective Function				
	Flow Time	Job Lateness	Idle Time	Job Tardiness	Jobs Late
500	1[illegible].5315	-13.7436	9.5547	13.2943	77
1000	20.856[illegible]	-12.4426	11.570[illegible]	18.4149	192
1500	20.5020	-11.8875	11.3103	19.4089	285
2000	20.7852	-12.1754	11.4467	20.1465	269

TABLE NO. 4

PERFORMANCE OF PRIORITY RULES

NL = 9 EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Flow Time Mean	Flow Time Standard Dev.	Job Lateness Mean	Job Lateness Std. Dev.	Idle Time Mean	Idle Time Std. Dev.	Job Tardiness Mean	Job Tardiness Std. Dev.	No. of Jobs late out of 1997
FDFS	SST	20.7852	18.4252	-12.1754	17.0836	11.4467	11.6453	20.1465	19.8099	369
	DUDT	22.0168	21.5677	-10.9890	14.6621	12.6680	14.7245	21.9930	24.7002	324
	MINSEQ	22.2458	19.7299	-10.7147	18.2306	12.9073	13.4948	24.5947	22.9677	449
	FIXSETSEQ ties by DUDT	22.4422	19.8601	-10.5184	18.0558	13.1045	13.4900	25.1948	22.4142	455
	FIXSETSEQ ties by SST	22.6271	20.0478	-10.3335	18.1837	13.2894	13.7970	25.6692	22.5780	455
	SPT	20.6060	18.1910	-12.3589	17.0368	11.2677	11.3353	19.2877	18.7345	344
	Slack/ Operation	22.6633	21.3119	-10.3151	15.5411	13.3206	14.7974	22.9113	23.5487	377
	LST	35.7743	172.5599	2.8242	169.0113	26.4505	171.0684	92.9672	356.2657	430
MAXQ	SST	20.4577	18.1303	-12.4966	17.3063	11.1210	11.3774	19.8193	19.8236	354
	DUDT	22.0457	21.7171	-10.9142	14.4354	12.7067	14.8529	23.2225	25.0732	316
	MINSEQ	22.1240	19.5691	-10.8409	18.3442	12.7858	13.2900	24.9919	22.1062	446
	FIXSETSEQ ties by DUDT	22.5125	19.9362	-10.4932	18.1196	13.1637	13.5864	24.9782	23.1992	448
	FIXSETSEQ ties by SST	22.2891	19.8693	-10.6300	18.0492	12.9606	13.5913	25.2912	23.3288	430
	SPT	20.5259	18.2119	-12.4462	17.2879	11.1850	11.4545	19.3526	18.9414	342
	Slack/ Operation	22.4736	21.1340	-10.5047	15.3969	13.1310	14.5402	21.4568	22.6368	368
	LST	35.5254	162.5100	2.5971	158.7748	26.2135	160.8484	99.6251	356.4018	377

TABLE NO. 5

PERFORMANCE OF PRIORITY RULES

NL = 8 EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Flow Time Mean	Flow Time Std. Dev.	Job Lateness Mean	Job Lateness Std. Dev.	Idle Time Mean	Idle Time Std. Dev.	Job Tardiness Mean	Job Tardiness Std. Dev.	No. of Jobs late out of 1997
FDFS	SST	21.9973	19.9943	-11.0085	16.9333	12.6465	13.2932	21.9513	22.2840	416
	DUDT	23.7118	23.6951	- 9.2939	14.0342	14.3631	16.8895	27.6017	28.7262	399
	MINSEQ	23.5743	21.2146	- 9.3848	18.0877	14.2372	15.0328	27.1353	24.5867	504
	FIXSETSEQ ties by DUDT	24.2283	21.7882	- 8.7439	18.3186	14.8874	15.6149	28.6376	25.3232	526
	FIXSETSEQ ties by SST	23.8672	21.4511	- 9.1049	18.2426	14.5264	15.3061	27.9082	25.0169	519
	SPT	21.5930	19.2876	-11.3720	16.9544	12.2547	12.5397	20.8438	21.2846	386
	Slack/ Operation	24.3329	23.2635	- 8.6708	15.0511	14.9861	16.7825	25.8057	26.8978	472
	LST	37.3726	176.7403	4.3666	173.2614	28.0368	175.1715	93.6242	360.7116	440
MAXQ	SST	21.5520	19.2811	-11.4537	16.9050	12.2032	12.4781	20.9707	20.8806	395
	DUDT	22.9630	22.5225	- 9.9969	14.4212	13.6240	15.6480	25.3733	26.7116	368
	MINSEQ	23.3252	21.0518	- 9.6868	18.1637	13.9746	14.8239	25.9354	24.5068	491
	FIXSETSEQ ties by DUDT	23.7359	21.1472	- 9.2184	18.0877	14.3992	14.9251	27.5758	24.5116	510
	FIXSETSEQ ties by SST	23.7694	21.2969	- 9.2064	18.2972	14.4289	15.1327	28.0267	24.6950	521
	SPT	21.5162	19.0100	-11.4895	16.8079	12.1674	12.1928	20.1547	19.7925	386
	Slack/ Operation	23.9858	22.0176	- 8.9863	15.2050	14.6450	16.5427	26.8897	26.4755	437
	LST	37.1547	189.7071	4.1029	185.3901	27.8116	188.0471	95.9354	395.5876	423

TABLE NO. 6

PERFORMANCE OF PRIORITY RULES

WL = 7 EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Flow Time Mean	Flow Time Std. Dev.	Job Lateness Mean	Job Lateness Std. Dev.	Idle Time Mean	Idle Time Std. Dev.	Job Tardiness Mean	Job Tardiness Std. Dev.	No. of Jobs late out of 1997
	SST	24.7794	23.1080	-8.2043	16.6928	15.4335	16.6168	28.1567	28.2024	541
	DUDT	26.4231	26.6660	-6.5827	13.7043	17.0743	19.8323	33.0636	31.8149	540
	MINSEQ	27.1034	24.7944	-5.8624	18.7891	17.7647	18.7934	33.5477	28.3627	646
	FIXSETSEQ ties by DUDT	27.6704	25.9573	-5.2618	19.2444	18.3396	19.9968	34.9053	30.5447	664
FDFS	FIXSETSEQ ties by SST	27.3476	25.2890	-5.6581	19.0026	17.9989	19.2850	33.4908	29.4358	660
	SPT	24.6757	22.7741	-8.2964	16.6708	15.3348	16.2377	27.6749	25.5441	543
	Slack/ Operation	27.4649	26.0865	-5.6113	15.2462	18.1039	19.6770	32.3920	29.3625	679
	LST	40.4735	198.521	7.4208	194.337	31.1303	196.921	84.2767	361.586	558
	SST	23.9453	21.8343	-9.0090	16.6961	14.6086	15.1500	25.1687	25.0716	497
	DUDT	25.9503	25.7725	-6.9818	13.9093	16.6195	18.9751	32.2485	30.5564	559
	MINSEQ	26.0961	23.8638	-6.8582	18.0869	16.7594	17.7005	30.2856	27.9587	634
MAXQ	FIXSETSEQ ties by DUDT	26.2777	24.1360	-6.6944	18.0182	16.9369	17.9258	32.0787	28.3257	614
	FIXSETSEQ ties by SST			N O T	T E S T E D					
	SPT	24.1888	21.9586	-8.8170	16.7391	14.8400	15.3229	25.5023	24.7856	522
	Slack/ Operation	26.8366	25.3831	-6.0956	14.6681	17.5058	18.8687	31.6001	28.7876	653
	LST	39.6789	168.101	6.6923	163.442	30.3524	166.236	82.0331	305.598	555

TABLE NO. 7

PERFORMANCE OF PRIORITY RULES

NL = 6 EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Flow Time Mean	Flow Time Std. Dev.	Job Lateness Mean	Job Lateness Std. Dev.	Idle Time Mean	Idle Time Std. Dev.	Job Tardiness Mean	Job Tardiness Std. Dev.	No. of Jobs late out of 1997
FDFS	SST	30.7984	31.3757	-2.1266	20.6368	21.4702	25.3348	39.3211	38.3384	814
	DUDT	34.2309	35.1017	1.3377	17.0028	24.9046	28.4873	45.7412	40.7454	966
	MINSEQ	33.3445	32.4711	0.3787	22.5729	24.0057	26.7432	43.0387	37.4576	886
	FIXSETSEQ ties by DUDT	33.7805	32.0658	0.7748	22.1289	24.4341	26.2069	43.5760	36.1547	913
	FIXSETSEQ ties by SST			NOT	TESTED					
	SPT	30.6611	31.0257	-2.3446	20.5358	21.3147	24.9697	38.9907	38.1789	794
	Slack/ Operation	35.2908	34.8486	2.3878	18.8919	25.9678	28.7565	45.2001	38.9042	1067
	LST	46.9739	180.255	13.929	174.491	37.6358	178.120	75.7880	263.612	864
MAXQ	SST	29.9866	30.4882	-2.9683	20.5375	20.6493	24.4249	38.5085	38.1103	775
	DUDT	30.6954	31.1971	-2.1867	15.1557	21.3757	24.4789	41.3276	36.7519	786
	MINSEQ	31.0333	29.5432	-1.9325	20.6224	21.6945	23.6947	39.8064	34.3513	814
	FIXSETSEQ ties by DUDT	32.0248	30.8714	-0.8860	20.1949	22.7003	24.9372	41.1939	35.6751	857
	FIXSETSEQ ties by SST			NOT	TESTED					
	SPT	30.1704	31.1923	-2.7945	21.4536	20.8321	25.2113	38.3876	38.8812	790
	Slack/ Operation	32.3521	31.5390	-0.5269	16.3848	23.0327	25.2703	40.9059	35.4783	938
	LST			NOT	TESTED					

TABLE NO. 8

INDIVIDUAL OPERATOR UTILISATION

NL = 9 EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Operator Number 1	2	3	4	5	6	7	8	9
FDFS	SST	32.26	65.02	35.51	38.74	66.05	55.99	8.35	47.62	69.24
	DUDT	31.78	65.94	36.99	38.63	65.58	55.06	8.19	47.58	69.14
	MINSEQ	29.20	65.01	36.95	38.35	65.12	55.50	7.33	46.84	68.36
	FIXSETSEQ ties by DUDT	30.52	64.80	37.47	38.05	65.45	54.96	7.29	47.21	68.58
	FIXSETSEQ ties by SST	29.96	64.86	37.60	38.23	64.70	55.32	7.88	47.32	68.30
	SPT	31.70	65.38	36.06	38.83	66.25	55.84	7.84	47.04	68.85
	Slack/ Operation	30.94	65.55	37.56	38.48	65.81	55.19	6.95	48.12	69.75
MAXQ	SST	28.94	65.78	36.50	38.44	67.14	56.02	6.86	48.27	69.95
	DUDT	29.39	65.81	35.99	38.55	66.58	55.72	7.49	48.75	69.43
	MINSEQ	29.43	65.00	36.13	38.31	65.70	55.18	6.49	47.40	68.98
	FIXSETSEQ ties by DUDT	29.35	65.57	35.15	38.22	65.81	56.01	8.54	47.16	68.31
	FIXSETSEQ ties by SST	28.79	64.90	36.34	38.35	65.68	56.06	7.58	57.69	68.89
	SPT	30.44	65.26	36.27	38.97	65.97	55.65	8.29	48.11	69.07
	Slack/ Operation	29.71	66.48	35.12	38.36	67.17	55.59	7.75	48.58	69.12

TABLE NO. 10

INDIVIDUAL OPERATOR UTILISATION

NL = 7 EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Operator Number 1	2	3	4	5	6	7
FDFS	SST	78.95	74.75	69.80	53.26	71.07	63.47	46.84
	DUDT	79.76	75.38	69.37	52.29	70.51	63.17	45.01
	MINSEQ	78.10	74.47	68.96	50.69	69.56	62.94	44.69
	FIXSETSEQ ties by DUDT	78.13	74.74	69.99	52.28	69.99	63.05	44.52
	FIXSETSEQ ties by SST	79.44	74.52	69.42	51.57	70.28	63.13	40.08
	SPT	79.30	74.39	69.41	53.45	70.86	63.54	45.82
	Slack/ Operation	79.79	75.09	68.70	51.74	70.63	63.86	46.09
MASQ	SST	79.87	75.02	68.69	50.26	71.99	62.37	43.88
	DUDT	79.85	74.04	69.54	52.36	70.79	63.65	44.32
	MINSEQ	78.85	74.13	68.00	50.58	70.51	61.62	42.13
	FIXSETSEQ ties by DUDT	78.79	74.57	69.22	50.72	69.45	62.32	42.71
	FIXSETSEQ ties by SST			NOT TESTED				
	SPT	79.94	74.74	69.27	52.42	70.76	63.23	44.47
	Slack/ Operation	80.35	75.08	69.69	51.89	71.32	61.91	43.39

TABLE NO. 11

INDIVIDUAL OPERATOR UTILISATION

NL = 6 EX = 8.00

Labor Assn. Rule	Job Assignment Rule	Operator Number 1	2	3	4	5	6
FDFS	SST	85.51	83.66	79.19	70.96	79.18	78.52
	DUDT	N O T	T E S T E D				
	MINSEQ	83.79	81.15	78.07	68.54	78.16	76.59
	FIXSETSEQ ties by DUDT	83.80	82.63	77.81	69.68	77.77	77.76
	FIXSETSEQ ties by SST	N O T	T E S T E D				
	SPT	85.95	83.61	78.86	71.44	78.92	78.83
	Slack/ Operation	85.33	83.41	79.65	71.01	79.16	79.83
MAXQ	SST	85.54	82.05	77.59	68.02	78.82	77.54
	DUDT	86.45	82.81	78.54	70.03	79.71	77.94
	MINSEQ	84.03	80.72	77.64	66.60	76.93	75.09
	FIXSETSEQ ties by DUDT	84.34	81.48	77.63	66.81	78.63	75.84
	FIXSETSEQ ties by SST	N O T	T E S T E D				
	SPT	85.98	82.75	80.13	69.18	79.78	78.69
	Slack/ Operation	85.39	82.16	79.02	69.32	79.40	76.60

TABLE NO. 12

OVERALL OPERATOR UTILISATION

EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Number of Operators (NL) 9	8	7	6
FDFS	SST	46.31	53.98	65.45	79.50
	DUDT	46.54	53.82	65.07	79.32
	MINSEQ	45.86	53.15	63.34	77.72
	FIXSETSEQ ties by DUDT	46.03	52.87	64.67	78.24
	FIXSETSEQ ties by SST	46.02	53.31	46.50	78.42
	SPT	46.40	53.74	65.25	79.60
	Slack/ Operation	46.48	53.90	65.13	79.73
MAXQ	SST	46.43	53.69	65.23	78.24
	DUDT	46.41	53.66	64.94	79.25
	MINSEQ	45.85	52.81	63.69	76.84
	FIXSETSEQ ties by DUDT	46.01	53.02	63.91	77.46
	FIXSETSEQ ties by SST	46.03	52.81	N O T	T E S T E D
	SPT	46.45	53.55	64.83	79.42
	Slack/ Operation	46.43	53.55	64.88	78.64

TABLE NO. 13

OVERALL MACHINE UTILISATION

EX = 3.00

Labor Assn. Rule	Job Assignment Rule	Number of Operators (NL)			
		9	8	7	6
FDFS	SST	46.31	47.99	50.90	53.00
	DUDT	46.54	47.94	50.61	53.00
	MINSEQ	45.86	47.25	49.93	51.81
	FIXSETSEQ ties by DUDT	46.03	47.30	50.30	52.16
	FIXSETSEQ ties by SST	46.01	47.38	50.16	-
	SPT	46.40	47.77	50.75	53.07
	Slack/ Operation	46.48	47.90	50.64	53.15
MAXQ	SST	46.43	47.73	50.22	52.16
	DUDT	46.41	47.67	50.51	52.83
	MINSEQ	45.85	48.05	49.54	51.22
	FIXSETSEQ ties by DUDT	46.01	47.13	49.75	51.64
	FIXSETSEQ ties by SST	46.03	46.94	N O T	T E S T E D
	SPT	46.45	47.60	50.40	52.95
	Slack/ Operation	46.43	47.38	50.46	52.43

TABLE NO. 14

PERFORMANCE OF PRIORITY RULES

EX = 2.00

Labor Assn. Rule	No. of Operators	Job Assignment Rule	Flow Time		Job Lateness		Idle Time		Job Tardiness		No. of Jobs late out of 1997
			Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	
FDFS	9	DUDT	40.03	41.50	10.17	19.34	33.72	34.60	54.25	43.75	1295
		MINSEQ	40.02	39.47	7.23	28.47	30.72	34.09	50.18	44.07	1094
	8	DUDT	80.71	67.55	48.03	47.97	71.44	61.55	91.92	65.86	1642
		MINSEQ	59.75	64.53	26.96	51.52	50.45	59.55	73.90	69.05	1381
	7	DUDT	NUMBER OF JOBS IN THE QUEUE OF MACHINE 3 ARE GREATER THAN 100								
		MINSEQ	152.0	191.3	120.2	180.0	142.9	187.5	164.2	194.2	1751

TABLE NO. 15

OVERALL OPERATOR AND MACHINE UTILIZATION

EX = 2.00

Labor Assn. Rule	Job Assignment Rule	Operation Utilisation			Machine utilisation		
		NL = 9	NL = 8	NL = 7	NL = 9	NL = 8	NL = 7
FDFS	DUDT	70.529	84.188	-	70.529	74.816	-
	MINSEQ	69.497	82.121	96.593	69.497	72.997	75.130

TABLE NO. 13

INDIVIDUAL OPERATOR UTILIZATION

EX = 2.00

Labor Assn. Rule	Number of Operators	Job Assignment Rule	Operator Number 1	2	3	4	5	6	7	8	9
	9	DUDT	62.28	85.22	68.74	57.89	86.34	79.73	33.87	71.15	89.54
		MINSEQ	62.34	82.33	64.76	57.51	84.59	77.67	31.34	68.56	87.38
FDFS	8	DUDT	93.81	89.51	83.44	73.63	89.73	86.11	72.59	84.52	-
		MINSEQ	91.10	88.58	81.58	72.11	88.11	84.46	69.36	81.67	-
	7	DUDT	NUMBER OF JOBS IN THE QUEUE OF MACHINE 3 ARE MORE THAN THAN 100								
		MINSEQ	98.07	98.16	97.20	94.26	96.05	96.55	95.86	-	-

APPENDIX - 2

COMPUTER PROGRAM

LISTING AND FLOW CHARTS

```
********************************************************************************************
                 *                                                                          *
                 *      M A I N   P R O G R A M                                             *
                 *                                                                          *
********************************************************************************************
IBJOB
IBFTC MAIN
      DIMENSION TMARVL(201),JOBSEQ(200,20),TMJOB(200,20),NMOP(200),
     1TMTOT(200),TMPLT(200),DUDT(200),TMSCOM(200),JINDEX(200),JLAST(9)
     2,JSTN(9,100),JOBOPR(9,100),TMARMC(9,200),JOBPR(9),NOPT(9)
     3,TMLSU(9),JCWAIT(9),JDWAIT(9),NQUE(9),ALPHA(9),
     4SETUP(10,10),WEFFCY(9,9),ISETCL(200),TMFNSH(9),TMSET1(10),
     5IPNDX(9),IOPER(9),P(200),C(200),R(200)
      DIMENSION DDUDT(200,20)
      DIMENSION T(9),UTLS(9),OU(9)
      DIMENSION SFTSEQ(11)
      COMMON /BLC1/TMARVL/BLC2/JSTN,JOBOPR/BLC3/JOBSEQ,TMJOB/BLC4/WEFFCY,
     1IPNDX/BLC5/JOBPR,NOPT,JLAST/BLC6/TMFNSH/BLC7/TMARMC/BLC8/TMSCOM/
     2BLC9/CLTIME/BLC10/TMTOT,TMPLT/BLC11/JINDEX/BLC12/NMOP/BLC13/P,C,R
     3/BLC14/ISETCL,SETUP/BLC15/DUDT/BLC16/N1,M/BLC17/IOPER/BLC18/EX
     4,ALPHA
     4/BLC19/TMLSU,DDUDT/BLC20/JCWAIT,JDWAIT,NQUE
      INTEGER SFTSEQ
      READ5,N,M1=(ALPHA(I),I=1,N),NL,EX
 5    FORMAT(I1,I3,9F3.1,I1,F4.1)
      DO11I=1,10
      DO12J=1,10
      X1=RNDY5(Y5)
      SETUP(I,J)=0.4*X1
      IF(I.EQ.J)SETUP(IYJ)=0.
 12   CONTINUE
      TMSET1(I)= .4*RNDY5(Y6)
 11   CONTINUE
      DO13I=1,N
      DO14J=1,N
      X1=RNDY5(Y7)
      IF(X1.LE.0.2)WEFFCY(I,J)=.0
      IF(X1.GT.0.2.AND.X1.LE.0.5)WEFFCY(I,J)=0.5+X1-0.2
      IF(X1.GT.0.5)WEFFCY(I,J)=X1
 14   CONTINUE
 13   CONTINUE
      REAL MEAN4=MEAN5,MEAN6
      REAL MEAN7
      REAL MEAN1=MEAN2,MEAN3
      INDEX1=1
      INDEX2=1
      EX=3.
2002  NL=9
 10   MT=0
```

```
      PRINT1601,NL,EX
1601  FORMAT(5X,*NL=*,I4,*EX=*,F10.4/5X,*--------                    *)
      PRINT1602,INDEX1,INDEX2
1602  FORMAT(5X,*INDEX1=*,I3,*INDEX2=*,I3)
      IX=3
      DO16I=1,10
      CALL RANDUM(IX,X1)
 16   CONTINUE
      ITAR11=0
      ITAR12=0
      CALL FLUN(30.00)
      ITAR13=0
      ITAR14=0
      IOL=0
      ITAR15=0
      PCOST=0.
      CCOST=0.
      ITAR16=0
      ITAR17=0
      ITAR18=0
      ITAR19=0
      ITAR20=0
      MT1=0
      MEAN7=0.
      SUM7=0.
      MEAN1=0.
      MEAN2=0.
      KLT=0
      MEAN3=0.
      SUM1=0.
      SUM2=0.
      SUM3=0.
      IJK=1
      MEAN4=0.
      MEAN5=0.
      MEAN6=0.
      SUM4=0.
      SUM5=0.
      SUM6=0.
      NOPTA=NL
      DO9I=1,200
 9    JINDEX(I)=
      DO2I=1,N
      T(I)=0.
      DO2J=1,N1
      JOBOPR(I,J)=0
 2    JSTN(I,J)=
      DO3I=1,N
      IOPER(I)=0
      JLAST(I)=0
      NOPT(I)=0
      TMFNSH(I)=-1.00
      TMLSD(I)=0.00
      JCWAIT(I)=0
```

```
      JDWAIT(I)=0
  3   NQUE(I)=0
      DO4I=1,NL
      OU(I)=0.
  4   IPNDX(I)=0
      TMARVL(1)= .00
      JOBNO=0
      CLTIME=0.0
      ICHANG=10
 111  IF(ICHANG.LT.10)GOTO100
      IF(IJK.GT.200)GOTO104
      JOBNO=IJK
      JINDEX(JOBNO)=1
      CALL DTGEN(JOBNO,IJK,IX)
C     NOW WE ARE ASSIGNING THE FIRST M/C FOR THIS OPERATOR
      DDUDT(JOBNO,1)=(DUDT(JOBNO)-CLTIME)/TMTOT(JOBNO)*TMJOB(JOBNO,1)+
     1CLTIME
      DO15J=2,20
      IF(JOBSEQ(JOBNO,J).EQ.0)GOTO15
      DDUDT(JOBNO,J)=(DUDT(JOBNO)-CLTIME)/TMTOT(JOBNO)*TMJOB(JOBNO,J)+
     1DDUDT(JOBNO,J-1)
  15  CONTINUE
      CALL ASFMC(JOBNO,NMC,M1,IOL)
C***WE WILL FIND WHETHER THIS M/C IS IDLE
      IF(TMFNSH(NMC).GTH0.00)GOTO107
      IF(NOPTA.EQ.0)GOTO108
      CALL LABASN(NMC,N,NL)
      IF(IOPER(NMC).GT.NL)GOTO108
      JOBPR(NMC)=JOBNO
      NOPT(NMC)=1
      NOPTA=NOPTA-1
      TMLSD(NMC)=-3.00
      IF(JLAST(NMC).EQ.0)GOTO109
      J1=JLAST(NMC)
      J2=JOBPR(NMC)
      J3=ISETCL(J1)
      J4=ISETCL(J2)
      T1=SETUP(J3,J4)
      J5=IOPER(NMC)
      TMFNSH(NMC)=TMARMC(NMC,J2)+TMJOB(J2,1)/WEFFCY(J5,NMC)+T1
      T(NMC)=T(NMC)+TMFNSH(NMC)-CLTIME
      JCWAIT(NMC)=JCWAIU(NMC)+1
      JDWAIT(NMC)=M1
      OU(J5)=OU(J5)+TMFNSH(NMC)-CLTIME
      GOTO110
 108  TMLSD(NMC)=TMARMC(NMC,JOBNO)
      TMFNSH(NMC)=-2.0
 107  NQUE(NMC)=NQUE(NMC)+1
      GOTO110
 109  J1=JOBPR(NMC)
      J2=ISETCL(J1)
      J5=IOPER(NMC)
      TMFNSH(NMC)=TMARM+(NMC,J1)+TMJOB(J1,1)/WEFFCY(J5,NMC)+TMSET1(J2)
      OU(J5)=OU(J5)+TMFNSH(NMC)-CLTIME
```

```
      T(NMC)=T(NMC)+TMFNSH(NMC)-CLTIME
      JCWAIT(NMC)=JCWAIO(NMC)+1
      JDWAIT(NMC)=M1
C*****FINDINF THE REASON OF NEXT SYSTEM CHANCE
 110  CALL SYSCHG(IJK,1+HANG,M,TMNXCG)
      CLTIME=TMN7CG
      GOTO111
 100  NMC=ICHANG
      TMLSD(NMC)=CLTIME
      JLAST(NMC)=JOBPR(NMC)
      K1=JDWAIT(NMC)
      NOPTA=NOPTA+1
      I1=IOPER(NMC)
      IPNDX(I1)=
      K2=JOBPR(NMC)
      JSTN(NMC,K1)=0
      NRMOP(K2)=NRMOP(K2)-1
      L1=JOBOPR(NMC,K1)
      L2=L1+1
      IOPER(NMC)=0
      TMPLT(K2)=3MPLT(K2)-TMJOB(K2,L1)
      P(K2)=(DUD3(K2)-C-TIME-TMPLT(K2))/(DUDT(K2)-CLTIME)
      TMFNSH(NMC)=-1.00
      IF(NRMOP(K2).EQ.0)GOTO116
      IF(JOBSEQ(K2,L2).EQ.0)GOTO116
 14   CALL MOVJOB(L2,L1YK2,NOPTA,TMSET1,NL,IOL,WEFFCY)
      IF(IOL.EQ.1)GOTO2001
      GOTO118
 116  CALL ANLYS(NMC,K2YMEAN1,MEAN2,MEAN3,MEAN4,MEAN5,MEAN6,MEAN7,SUM1,
     1SUM2,SUM3,2UM4,SUM5,SUM6,SUM7,MT,MT1,ITAR11,ITAR12,ITAR13,ITAR14,
     2ITAR15,ITAR16,ITAR17,ITAR18,ITAR19,ITAR20,PCOST,CCOST)
 118  CALL MCASN(NL,NOPTA,TMSET1,SLACK,INDEX1,INDEX2,T,OU)
 117  IF(MT.GE.2000)GOTO125
      IF(MT.EQ.500)GOTO125
      IF(MT.FQ.1000)GOTO125
      IF(MT.EQ.1500)GOTO125
      GOTO110
 125  MT2=MT-KLT
      AM1=MEAN1/FLOAT(M02-50)
      AM2=MEAN2/FLOAT(MT2-50)
      AM3=MEAN3/FLOAT(M02-50)
      A1=(SUM1-FLOAT(MT2-50)*AM1**2)/FLOAT(MT2-51).
      A2=(SUM2-FLOAT(MT2-50)*AM2**2)/FLOAT(MT2-51)
      A3=(SUM3-FLOAT(MT2-50)*AM2**2)/FLOAT(MT2-50-1)
      STD1=SQRT(ABS(A1))
      STD2=SQRT(ABS(A2))
      STD3=SQRT(ABS(A3))
      PRINT1005
 1005 FORMAT(5X,*STATISOICS OF JOBS ,IGNORING FIRST 50 JOBS ARE AS
     1FOLLOWS*)
      PRINT1001,AM1,STD1
 1001 FORMAT(//57,*STATISTICS OF TIME FOR WHICH JOB WAS IN THW SHOP ARE
     1AS FOLLOWS*/5X,*MEAN VALUE =*,F20.4/5X,*STANDARD DEVIATION =*,F20.
```

```
     24)
      PRINT1002,AM2,STD2
 1002 FORMAT(//57,*STATISTICS FOR PENALTY TIMES ARE AS FOLLOWS*/5X,*
     1MEAN VALUE =*,F20H4/5X,*STANDARD DEVIATION =*,F20.4)
      PRINT1003,AM3,STD
 1003 FORMAT(//57,*STATISTICS   FOR  IDLE  TIME  ARE  A2  FOLLOWS*/5
     1X,*MEAN VALUE =*,F20.4/5X,*STANDARD DEVIATION =*,F20.4)
      AM1=MEAN4/FLOAT(M02-100)
      AM2=MEAN5/FLOAT(M02-100)
      AM3=MEAN6/FLOAT(M02-100)
      A1=(SUM4-FLOAT(MT2-100)*AM1**2)/FLOAT(MT2-101)
      A2=(SUM5-FLOAT(MT2-100)*AM2**2)/FLOAT(MT2-101)
      A3=(SUM6-FLOAT(MT2-100)*AM3**2)/FLOAT(MT2-101)
      STD1=SQRT(ABS(A1))
      STD2=SQRT(ABS(A2))
      STD3=SQRT(ABS(A3))
      PRINT1006
 1006 FORMAT(5X,*STATISTICS OF JOBS ,IGNORING FIRST 100 JOBS ARE AS
     1FOLLOWS*)
      PRINT1001,AM1,STD1
      PRINT1002,AM2,STD2
      PRINT1003,AM3,STD3
      MT=MT+1
      AM4=MEAN7/FLOAT(MT1)
      A4=(SUM7-FLOAT(MT1)*AM4**2)/FLOAT(MT1-1)
      STD4=SQRT(ABS(A4))
      PRINT1012,MT2,MT1YAM4,STD4
 1012 FORMAT(5X,*  NUMBRE OF JOBS WITH POSITIVE LATENESS , OUT OF TOTAL
     1JOBS*,I5,* ARE*,I5/5X,*AVERAGE T TARDINESS*,F10.4/5X,*STD. DEVIA.*
     2,F10.4)
      KLT=KLT+1
      IF(MT.LT.2 00)GOTO110
      PRINT1011,CLTIME
 1011 FORMAT(5X,*CLOCK UIME =*,F20.4)
      KMP=0
      DO303I=1,N
      DO303II=1,N1
      J1=JSTN(I,II)
      IF(J1.EQ.0)GOTO303
      KMP=KMP+1
 303  CONTINUE
      PRINT1008,KMP
 1008 FORMAT(5X,*NUMBER OF JOBS WAITING=*,I4)
      DO301I=1,N
      IF(TMFNSH(I).LT.UH)GOTO301
      T(I)=T(I)-3MFNSH(I)+CLTIML
      J1=IOPER(I)
      OU(J1)=OU(J1)-TMFNSH(I)+CLTIME
 301  CONTINUE
      PRINT311,PCOST,CCOST
 311  FORMAT(5X,*P.COST8*,F20.4/5X,*C.COST=*,F20.4)
      DO306I=1,N
```

```
 306   UTLS(I)=T(I)/CLTIML*100.
       PRINT307,(4TLS(I)YI=1,N)
 307   FORMAT(5X,*M/C UTILISATIONIS AS FOLLOWS*/5X,9F8.3)
       PRINT308,ITAR11,ITAR12,ITAR13,ITAR14,ITAR15,ITAR16,ITAR17,ITAR18,
      1ITAR19,ITAR20
 308   FORMAT(5X,*TARLINGS HISTOGRAM IS*/5X,10I5)
       DO313I=1,NL
 313   OU(I)=OU(I)/CLTIME*100.
       PRINT314,(OU(I),I31,NL)
 314   FORMAT(5X,*OPERATOR UTILISATION IS*/5X,9F8.3)
 2003  NL=NL-1
       IF(NL.GT.5)GOTO10
       INDEX1=INDEX1+1
       IF(INDEX1.LT.5)GOTO2002
       INDEX2=INDEX2+1
       IF(INDEX2.LT.3)GO002002
 2001  EX=EX+1.
       IF(EX.LT.1 .)GOTO2002
       GOTO115
 104   PRINT1101
 1101  FORMAT(//5X,*100 AS MAXIMUM NUMBER OF JOBS IS NOT SUFFICIENT*)
       GOTO2001
 115   STOP
       END
C*****************************************************************************
*                *                                                          *
*                *     S U B R O U T I N E   M C A S N                      *
*                *                                                          *
C*****************************************************************************
$IBFTC SUB2     DECK
      SUBROUTINE MCASN(NL,NOPTA,TMSET1,ZLACK,INDEX1,INDEX2,T,OU)
      DIMENSION TMARVL(201),JOBSEQ(200,20),TMJOB(200,20),NPMOP(200),
     1TMTOT(200)=TMPLT(200),DUDT(200),TMSCOM(200),JINDEX(200),JLAST(9)
     2,JSTN(9,100),JOBOPR(9,100),TMARNC(9,200),JOBPR(9),NOPT(9)
     3,TMLSD(9),JCWAIT(9),JDWAIT(9),NQUE(9),ALPHA(9),
     4SETUP(10,1 ),WEFF+Y(9,9),ISETCL(200),TMFNSH(9),TMSET1(10),
     5IPNDX(9),ICPER(9)YP(200),C(200),R(200)
      DIMENSION DDUDT(200,20)
      DIMENSION 3(9),OU(9)
      DIMENSION X(9),KB(9)
      COMMON/BLC1/TMARV-/BLC2/JSTN,JOBOPR/BLC3/JOBSEQ,TMJOB/BLC4/WEFFCY,
     1IPNDX/BLC51JOBPR,NOPT,JLAST/BLC6/TMFNSH/BLC7/TMARNC/BLC8/TMSCOM/
     2BLC9/CLTIME/BLC10/TMTOT,TMPLT/BLC11/JINDEX/BLC12/NPMOP/BLC13/P,C,R
     3/BLC14/ISE3CL,SETUP/BLC15/DUDT/BLC16/MIN/BLC17/IOPER/BLC18/EX
     4,ALPHA
     4/BLC19/TML2D,DDUDD/BLC20/JCWAIT,JDWAIT,NQUE
      INTEGER SE3SEQ
      DO123IP=1,N
      IF(NOPTA.EQ.0)GOTO124
      KJ=0
      DO119I=1,N
```

```
      KH=0
      IF(TMFNSH(I).GE.0H)GOTO119
      IF(TMLSD(I).LT.0.)GOTO119
      IF(NQUE(I).EQ.0)GOTO119
      DO120J=1,NL
      IF(IPNDX(J).EQ.(-4))GOTO120
      IF(WEFFCY(J,I).EQ.0.)GOTO120
      KH=KH+1
      GOTO121
120   CONTINUE
      IF(KH.EQ.0)GOTO119
121   KJ=KJ+1
      GOTO(1,2),INDEX2
 1    X(KJ)=TMLSD(I)
      GOTO6
 2    X(KJ)=-NQUE(I)
 6    KB(KJ)=I
119   CONTINUE
      IF(KJ.EQ.0)GOTO124
      CALL MIN(X=KJ,KA)
      NMC=KB(KA)
      TMLSD(NMC)=-3.00
      CALL LABASN(NMC,NYNL)
      CALL JOBASN(NMC,JOB1,JL,N1,INDEX1)
      NOPTA=NOPTA-1
      JOBPR(NMC)=JOB1
      JCWAIT(NMC)=JCWAIU(NMC)+1
      JDWAIT(NMC)=JL
      NQUE(NMC)=NQUE(NM+)-1
      J1=JLAST(NMC)
      IF(JLAST(NMC).EQ.0)GOTO122
      J2=JOBPR(NMC)
      J3=ISETCL(J1)
      J4=ISETCL(J2)
      T1=SETUP(J3,J4)
      NOPT(NMC)=JOBOPR(NMC,JL)
      J5=NOPT(NMC)
      J6=IOPER(NMC)
      TMFNSH(NMC)=CLTIME+T1+TMJOB(J2,J5)/WEFFCY(J6,NMC)
      OU(J6)=OU(J6)+TMFNSH(NMC)-CLTIME
      T(NMC)=T(NMC)+TMFNSH(NMC)-CLTIME
      GOTO123
122   J2=JOBPR(NMC)
      J3=ISETCL(J2)
      J4=JOBOPR(NMC,JL)
      NOPT(NMC)=J4
      J5=IOPER(NMC)
      TMFNSH(NMC)=CLTIME+TMSET1(J3)+TMJOB(J2,J4)/WEFFCY(J5,NMC)
      T(NMC)=T(NMC)+TMFNSH(NMC)-CLTIME
      OU(J5)=OU(J5)+TMFNSH(NMC)-CLTIME
 123   CONTINUE
124   RETURN
```

```
      END
C***************************************************************************************
*                 *                                                                    *
*                 *     S U B R O U T I N E   D T G E N                                *
*                 *                                                                    *
C***************************************************************************************
$IBFTC SUB1     DECK
      SUBROUTINE DTGEN(JOBNO,IJK,IX)
      DIMENSION TMARVL(201),JOBSEQ(200,20),TMJOB(200,20),NRMOP(200),
     1TMTOT(200),TMPLT(200),DUDT(200),ALPHA(9),JINDEX(200),ISETCL(200),
     2SETUP(10,10),P(200),C(200),R(200)
      COMMON/BLC1/TMARVL/BLC3/JOBSEQ,TMJOB/BLC10/TMTOT,TMPLT/BLC12/NRMOP
     1/BLC13/P,C,R/BLC14/ISETCL,SETUP/BLC15/DUDT/BLC16/N1,N/
     2BLC11/JINDEX/BLC18/EX,ALPHA
      J1=JOBNO
 106  IF(J1.EQ.200)GOTO101
      J1=J1+1
      GOTO102
 101  J1=1
      IF(JINDEX(J1).EQ.0)GOTO103
 105  J1=J1+1
      IF(JINDEX(J1).EQ.0)GOTO103
      GOTO105
 102  IF(JINDEX(J1).EQ.0)GOTO103
      GOTO106
 103  IJK=J1
      CALL RANNUM(IX,X1)
      TMARVL(IJK)=TMARVL(JOBNO)+(-EX)*ALOG(X1)
      I=0
      NOP=0
  2   I=I+1
  4   CALL RANNUM(IX,X1)
      IF(X1.LE.0.1)JOBSEQ(JOBNO,I)=1
      IF(X1.GT.0.1.AND.X1.LE.0.2)JOBSEQ(JOBNO,I)=2
      IF(X1.GT.0.2.AND.X1.LE.0.3)JOBSEQ(JOBNO,I)=3
      IF(X1.GT.0.3.AND.X1.LE.0.4)JOBSEQ(JOBNO,I)=4
      IF(X1.GT.0.4.AND.X1.LE.0.5)JOBSEQ(JOBNO,I)=5
      IF(X1.GT.0.5.AND.X1.LE.0.6)JOBSEQ(JOBNO,I)=6
      IF(X1.GT.0.6.AND.X1.LE.0.7)JOBSEQ(JOBNO,I)=7
      IF(X1.GT.0.7.AND.X1.LE.0.8)JOBSEQ(JOBNO,I)=8
      IF(X1.GT.0.8.AND.X1.LE.0.9)JOBSEQ(JOBNO,I)=9
      IF(X1.GT.0.9)GOTO1
      IF(I.EQ.1)GOTO3
      IF(JOBSEQ(JOBNO,I).EQ.JOBSEQ(JOBNO,I-1))GOTO4
  3   NOP=NOP+1
      IF(I.LT.20)GOTO2
  1   IF(NOP.EQ.0)GOTO4
      NRMOP(JOBNO)=NOP
      DO5I=1,NOP
      M=JOBSEQ(JOBNO,I)
      CALL RANNUM(IX,X1)
      TMJOB(JOBNO,I)=-ALPHA(M)*ALOG(X1)
  5   CONTINUE
```

```
      IF(NOP.EQ.20)GOTO6
      N2=NOP+1
      DO7I=N2,20
      TMJOB(JOBNO,I)=0.00
  7   JOBSEQ(JOBNO,I)=0
  6   T1=0.0
      DO8I=1,NOP
  8   T1=T1+TMJOB(JOBNOYI)
      CALL RANNUM(IX,X1)
      X2=(2.+3.*X1)*T1+TMARVL(JOBNO)
      DUDT(JOBNO)=X2
      CALL RANNUM(IX,X1)
      IF(X1.LE.0.1)ISETCL(JOBNO)=1
      IF(X1.GT.0.1.AND.X1.LE.0.2)ISETCL(JOBNO)=2
      IF(X1.GT.0.2.AND.X1.LE.0.3)ISETCL(JOBNO)=3
      IF(X1.GT.0.3.AND.X1.LE.0.4)ISETCL(JOBNO)=4
      IF(X1.GT.0.4.AND.X1.LE.0.5)ISETCL(JOBNO)=5
      IF(X1.GT.0.5.AND.X1.LE.0.6)ISETCL(JOBNO)=6
      IF(X1.GT.0.6.AND.X1.LE.0.7)ISETCL(JOBNO)=7
      IF(X1.GT.0.7.AND.X1.LE.0.8)ISETCL(JOBNO)=8
      IF(X1.GT.0.8.AND.X1.LE.0.9)ISETCL(JOBNO)=9
      IF(X1.GT.0.9)ISETCL(JOBNO)=10
      TMTOT(JOBNO)=T1
      TMPLT(JOBNO)=T1
      RETURN
      END
C************************************************************************************
*                *
*                *    S U B R O U T I N E   A S F M C
*                *
C************************************************************************************
$IBFTC SUB12    DECK
      SUBROUTINE ASFMC(JOBNO,NMC,M1,IOL)
      DIMENSION JSTN(9,100),JOBSEQ(200,20),JOBOPP(9,100),TMARMC(9,200)
     1,TMARVL(201),TMJOB(200,20)
      COMMON/BLC2/JSTN,JOBOPP/BLC1/TMARVL/BLC3/JOBSEQ,TMJOB/BLC7/TMARMC
     1BLC16/N1,N
      NMC=JOBSEQ(JOBNO,1)
      DO1I=1,N1
      M1=I
      IF(JSTN(NMC,I).EQ.0)GOTO2
  1   CONTINUE
      IOL=1
      PRINT100,NMC
 100  FORMAT(//5X,*NUMBER OF JOBS WAITING ON M/C*,I1,*ARE MORE THAN 100
     1*)
      RETURN
  2   JSTN(NMC,M1)=JOBNO
      JOBOPP(NMC,M1)=1
      TMARMC(NMC,JOBNO)=TMARVL(JOBNO)
      RETURN
      END
C************************************************************************************
```

```
C*****************************************************************************
*                   *
*                   *     S U B R O U T I N E   S Y S C H G
*                   *
C*****************************************************************************
$IBFTC SUB13    DECK
      SUBROUTINE SYSCHG(IJOB,ICHANG,N,TMNXCG)
      COMMON/BLC1/TMARVL/BLC6/TMFNSH
      DIMENSION TMARVL(201),TMFNSH(9)
      ICHANG=10
      TM1=TMARVL(IJOB)
      TMNXCG=TM1
      DO1I=1,N
      IF(TMFNSH(I).LT.0.00)GOTO1
      IF(TMFNSH(I).GT.TMNXCG)GOTO1
      ICHANG=I
      TMNXCG=TMFNSH(I)
    1 CONTINUE
      RETURN
      END
C*****************************************************************************
*                   *
*                   *     S U B R O U T I N E   M O V J O B
*                   *
C*****************************************************************************
$IBFTC SUB14    DECK
      SUBROUTINE MOVJOB(L2,L1,K2,NOPTA,TMSET1,N1,IOL,WEFFCY)
      COMMON/BLC2/JSTN,JOBOPR/BLC3/JOBSEQ,TMJOB/BLC5/JOBPR,NOPT,JLAST/
     1BLC6/TMFNSH/BLC7/TMARMC/BLC9/CLTIME/BLC14/ISETCL,SETUP
     2/BLC17/IOPER/BLC20/JCWAIT,JDWAIT,NQUE/BLC19/TMLSD
      COMMON/BLC16/N1,N
      DIMENSION JSTN(9,100),JOBOPR(9,100),TMFNSH(9),JOBSEQ(200,20),
     1TMJOB(200,20),NPMOP(200),TMARMC(9,200),NOPT(9),JOBPR(9),IOPER(9)
     2,JLAST(9),ISETCL(200),SETUP(10,10),JCWAIT(9),JDWAIT(9),NQUE(9)
     3,TMLSD(9),TMSET1(10)
      DIMENSION WEFFCY(9,9)
      NMC=JOBSEQ(K2,L2)
      DO1I=1,N1
      M2=I
      IF(JSTN(NMC,I).EQ.0)GOTO2
    1 CONTINUF
      GOTO3
    2 JSTN(NMC,M2)=K2
      TMARMC(NMC,K2)=CLTIME
      JOBOPR(NMC,M2)=L2
      NQUE(NMC)=NQUE(NMC)+1
      GOTO7
    3 IOL=1
      PRINT100,NMC
  100 FORMAT(//5X,*THE NO. OF JOBS WAITING ON M/C*,I1,*GT100*)
    7 RETURN
      END
C*****************************************************************************
```

```
C*****************************************************************************
*                 *
*                 *     S U B R O U T I N E   L A D A S N
*                 *
C*****************************************************************************
$IBFTC SUB15    DECK
      SUBROUTINE LADASN(NMC,N,NL)
      COMMON/BLC4/WEFFCY,IPNDX/BLC17/IOPER
      DIMENSION WEFFCY(9,9),IOPER(9),X(9),IPNDX(9),IK(9)
      J=0
      DO1I=1,NL
      IF(IPNDX(I).EQ.(-4))GOTO1
      IF(WEFFCY(I,NMC).EQ.0.00)GOTO1
      J=J+1
      IK(J)=I
      X(J)=-WEFFCY(I,NMC)
   1  CONTINUE
      IF(J.EQ.0)GOTO2
      CALL MIN(X,J,K)
      L=IK(K)
      IOPER(NMC)=L
      IPNDX(L)=-4
      RETURN
    2 IOPER(NMC)=NL+1
      RETURN
      END
C*****************************************************************************
*                 *
*                 *     S U B R O U T I N E   J O B A S N
*                 *
C*****************************************************************************
$IBFTC SUB16
      SUBROUTINE JOBASN(NMC,JOB1,JL,N1,INDEX1,TMSET1)
      COMMON/BLC5/JOBPR,NOPT,JLAST/BLC14/ISETCL,SETUP
      COMMON/BLC2/JSTN,JOBOPR/BLC13/P,C,R/BLC19/TMLSD,DDUDT/BLC9/CLTIME
     1/BLC3/JOBSEQ,TMJOB/BLC15/DUDT
      DIMENSION DUDT(200)
      DIMENSION JOBSEQ(200,20),TMJOB(200,20)
      DIMENSION DDUDT(200,20),TMLSD(200)
      DIMENSION JSTN(9,100),JOBOPR(9,100),P(200),C(200),R(200),X(100),
     1KB(100)
      DIMENSION JOBPR(9),NOPT(9),JLAST(9),ISETCL(200),SETUP(10,10),
     1TMSET1(10)
      KJ=0
   2  DO3I=1,N1
      J1=JSTN(NMC,I)
      IF(J1.EQ.0)GOTO3
      J2=JOBOPR(NMC,I)
      KJ=KJ+1
      KB(KJ)=I
      J3=JLAST(NMC)
      IF(J3.EQ.0)GOTO6
      J4=ISETCL(J3)
```

```
      J5=ISETCL(J1)
      X(KJ)=TMJOB(J1,J2)+SETUP(J4,J5)
      GOTO3
    6 J4=ISETCL(J1)
      X(KJ)=TMJOB(J1,J2)+TMSFT1(J4)
   3  CONTINUE
      CALL MIN(X,KJ,KA)
      JL=KB(KA)
      JOB1=JSTN(NMC,JL)
      RETURN
      END
C**************************************************************************************
*                   *
*                   *     S U B R O U T I N E   A N L Y S
*                   *
C**************************************************************************************
$IBFTC SUB17    DECK
      SUBROUTINE ANLYS(NMC,K2,MEAN1,MEAN2,MEAN3,MEAN4,MEAN5,MEAN6,MEAN7
     1SUM1,SUM2,SUM3,SUM4,SUM5,SUM6,SUM7,MT,MT1,ITAR11,ITAR12,ITAR13,
     3ITAR14,ITAR15,ITAR16,ITAR17,ITAR18,ITAR19,ITAR20,PCOST,CCOST)
      DIMENSION JINDEX(200),TMSCOM(200),DUDT(200),TMARVL(201),TMTOT(200
     1,TMPLT(200),TMARMC(9,200)
      COMMON/BLC1/TMARVL/BLC7/TMARMC/BLC15/DUDT/BLC8/TMSCOM/BLC10/TMTOT
     1TMPLT/BLC11/JINDEX/BLC9/CLTIME/BLC13/P,C,R
      DIMENSION P(200),C(200),R(200)
      REAL MEAN7
      REAL MEAN1=MEAN2,MEAN3
      REAL MEAN4=MEAN5,MEAN6
      TMSCOM(K2)=CLTIME
      T1=TMSCOM(K2)-TMARVL(K2)
      T2=TMSCOM(K2)-DUDJ(K2)
      T3=T1-TMTO3(K2)
      IF(T2.GT.0.)GOTO3
      GOTO2
  3   MT1=MT1+1
      MEAN7=MEAN7+T1
      SUM7=SUM7+T1**2
  2   JINDEX(K2)=0
      IF(T2.LT.(-20.))ITAR11=ITAR11+1
      IF(T2.GE.(-20.).AND.T2.LT.(-15.))ITAR12=ITAR12+1
      IF(T2.GE.(-15.).AND.T2.LT.(-10.))ITAR13=ITAR13+1
      IF(T2.GE.(-10.).AND.T2.LT.(-5.))ITAR14=ITAR14+1
      IF(T2.GE.(-5.).AND.T2.LT.0.)ITAR15=ITAR15+1
      IF(T2.GE.0.0.AND.T2.LT.5.0)ITAR16=ITAR16+1
      IF(T2.GE.5.0.AND.T2.LT.10.0)ITAR17=ITAR17+1
      IF(T2.GE.10.0.AND.T2.LT.15.0)ITAR18=ITAR18+1
      IF(T2.GE.15.0.AND.T2.LT.20.0)ITAR19=ITAR19+1
      IF(T2.GE.20.0)ITAR20=ITAR20+1
      MT=MT+1
      IF(MT.LT.5 )GOTO1
      MEAN1=MEAN1+T1
      MEAN2=MEAN2+T2
```

```
      MEAN3=MEAN3+T3
      SUM1=SUM1+T1**2
      SUM2=SUM2+T2**2
      SUM3=SUM3+T3**2
      IF(MT.LE.100)GOTO1
      MEAN4=MEAN4+T1
      MEAN5=MEAN5+T2
      MEAN6=MEAN6+T3
      SUM4=SUM4+31**2
      SUM5=SUM5+T2**2
      SUM6=SUM6+33**2
  1   RETURN
      END
C****************************************************************************************************
*                    *                                                                              *
*                    *       S U B R O U T I N E   M I N                                            *
*                    *                                                                              *
C****************************************************************************************************
$IBFTC SUB18    DECK
      SUBROUTINE MIN(X,J,K)
      DIMENSION X(200)
      K=1
      SN=X(1)
      IF(J.EQ.1)GOTO2
      DO1I=2,J
      IF(X(I).GT.SN)GOTO1
      K=I
      SN=X(I)
   1  CONTINUE
  2   RETURN
      END
C****************************************************************************************************
*                    *                                                                              *
*                    *       S U B R O U T I N E   R A N N U M                                      *
*                    *                                                                              *
C****************************************************************************************************
$IBFTC SUB19    DECK
      SUBROUTINE RANNUM(IX,X1)
      IX=IX*3125
      X1=FLOAT(IX)/34359738367.D0
      RETURN
      END
$ENTRY
```

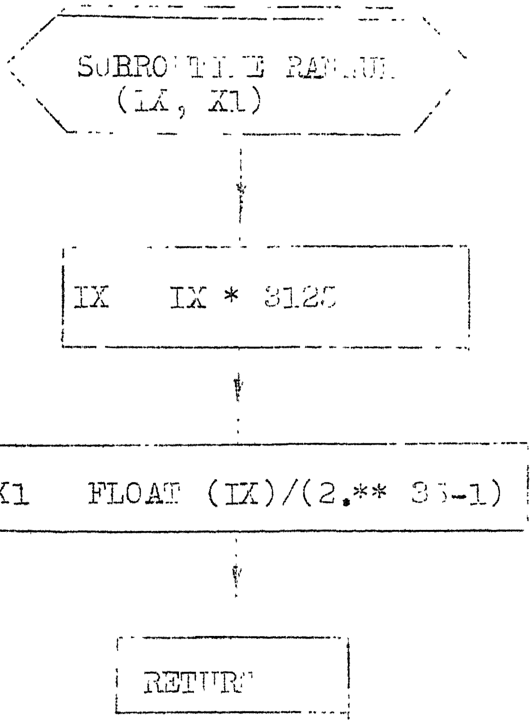


Fig. 5. Flow Chart for RANNUM

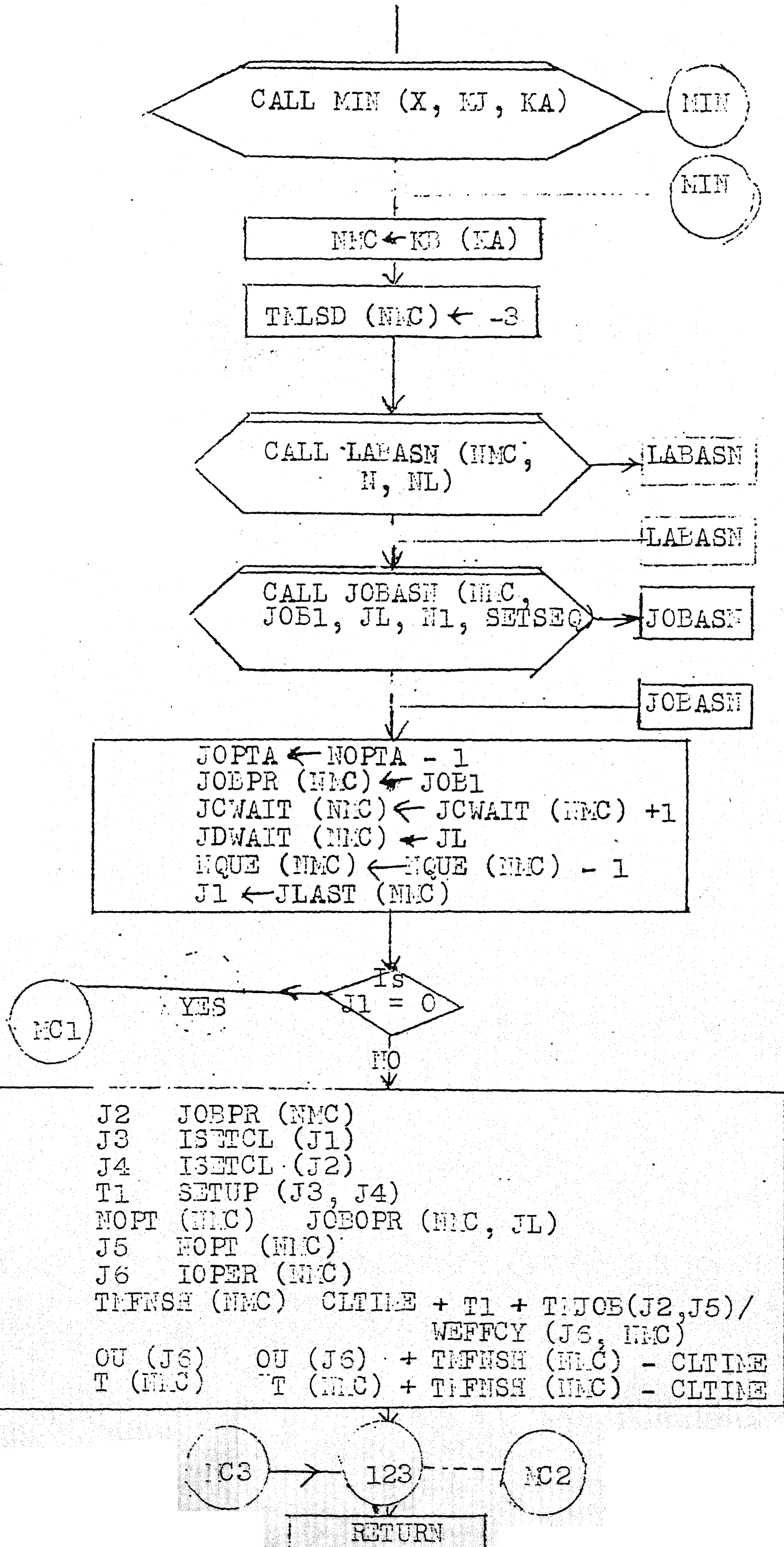


Fig. 12. (Continued)

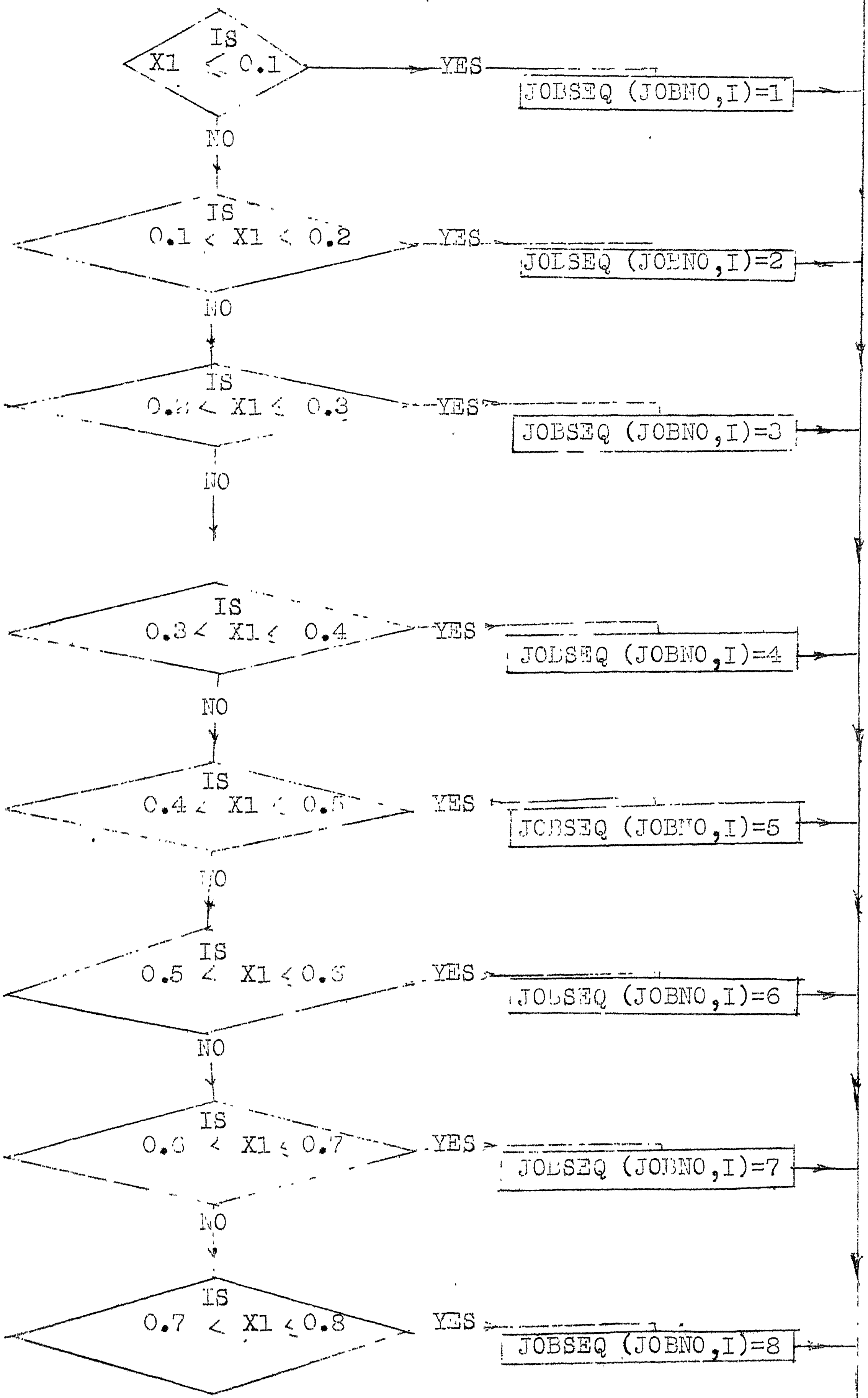


Fig. 6. (Continued)

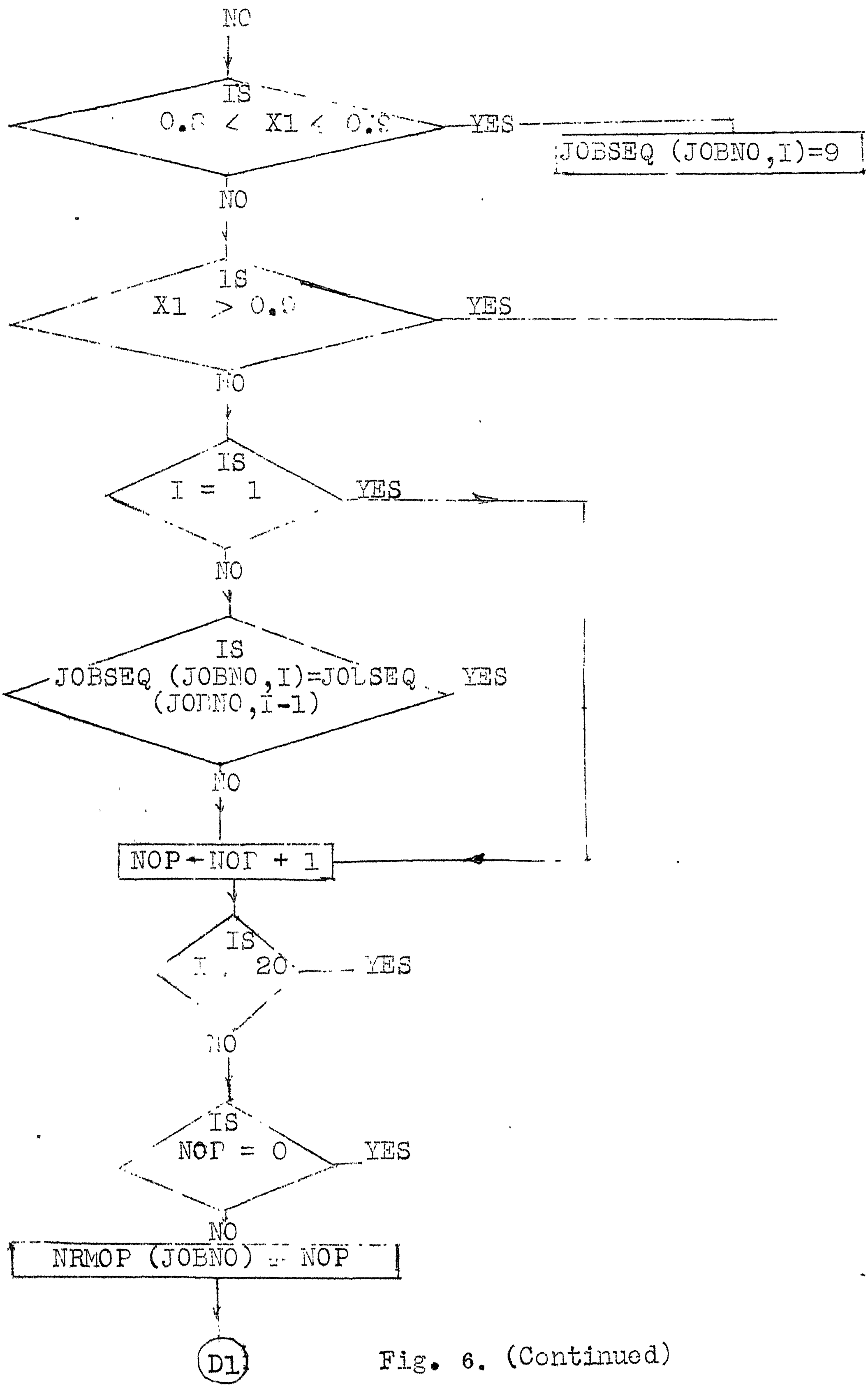


Fig. 6. (Continued)

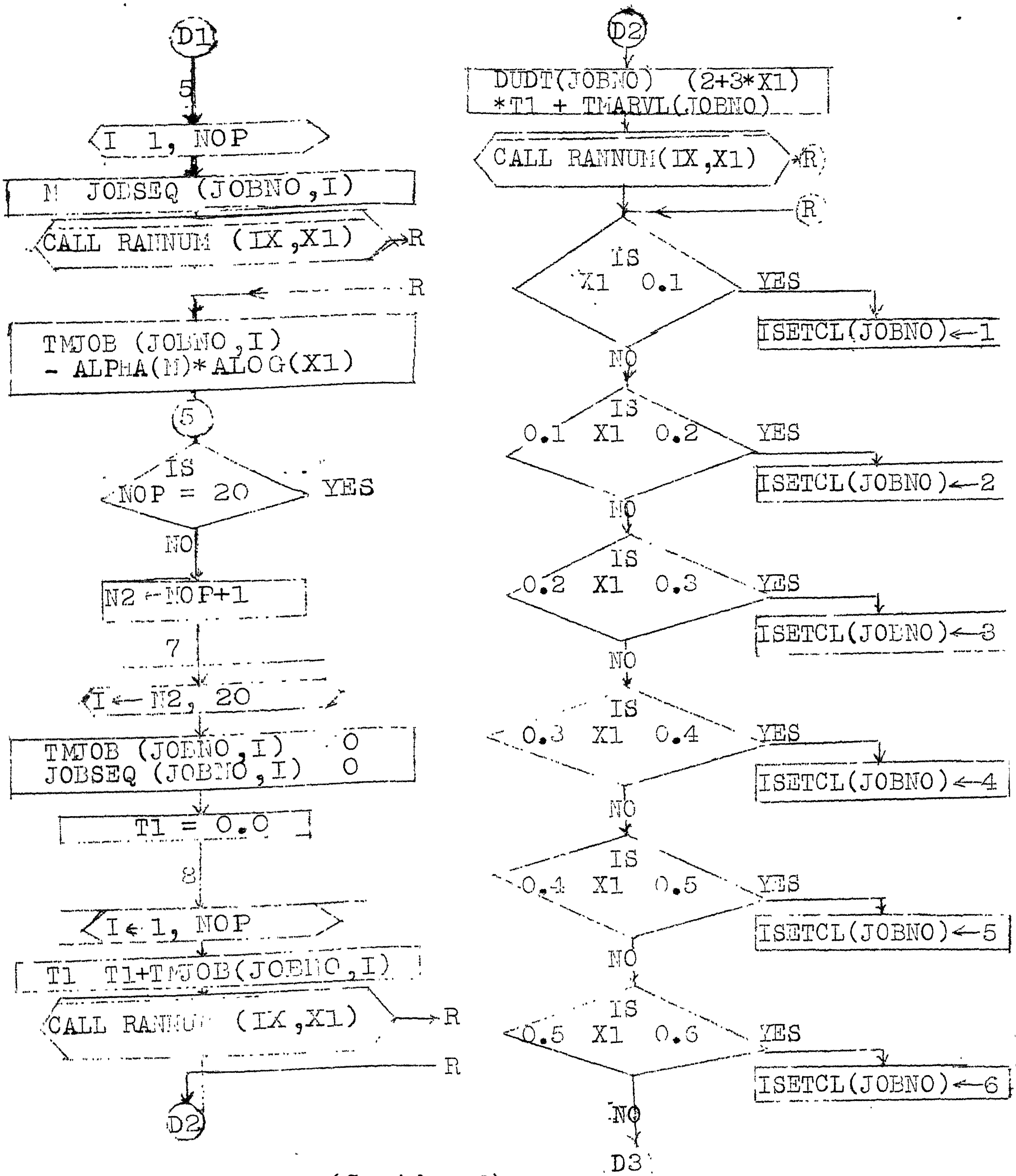


Fig. 6. (Continued)

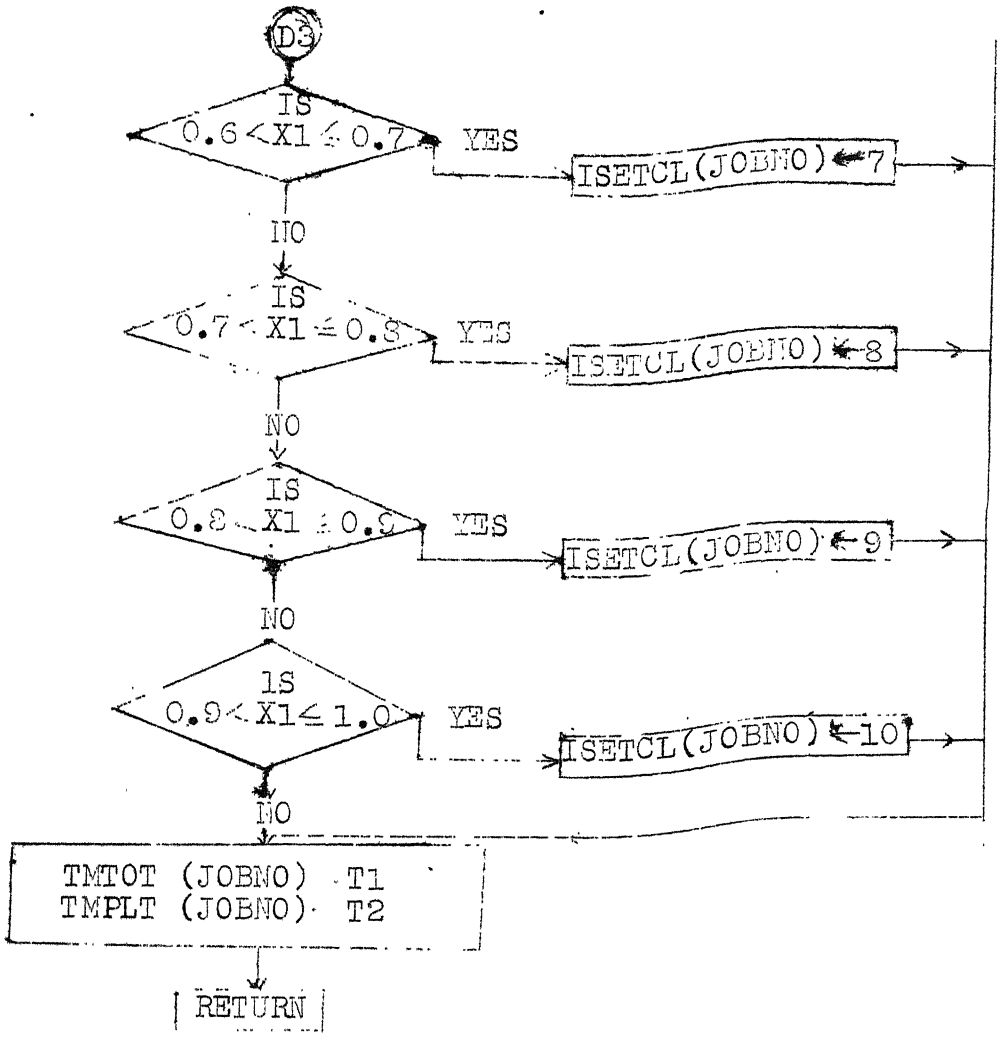


Fig. 6. Flow Chart for DTGEN

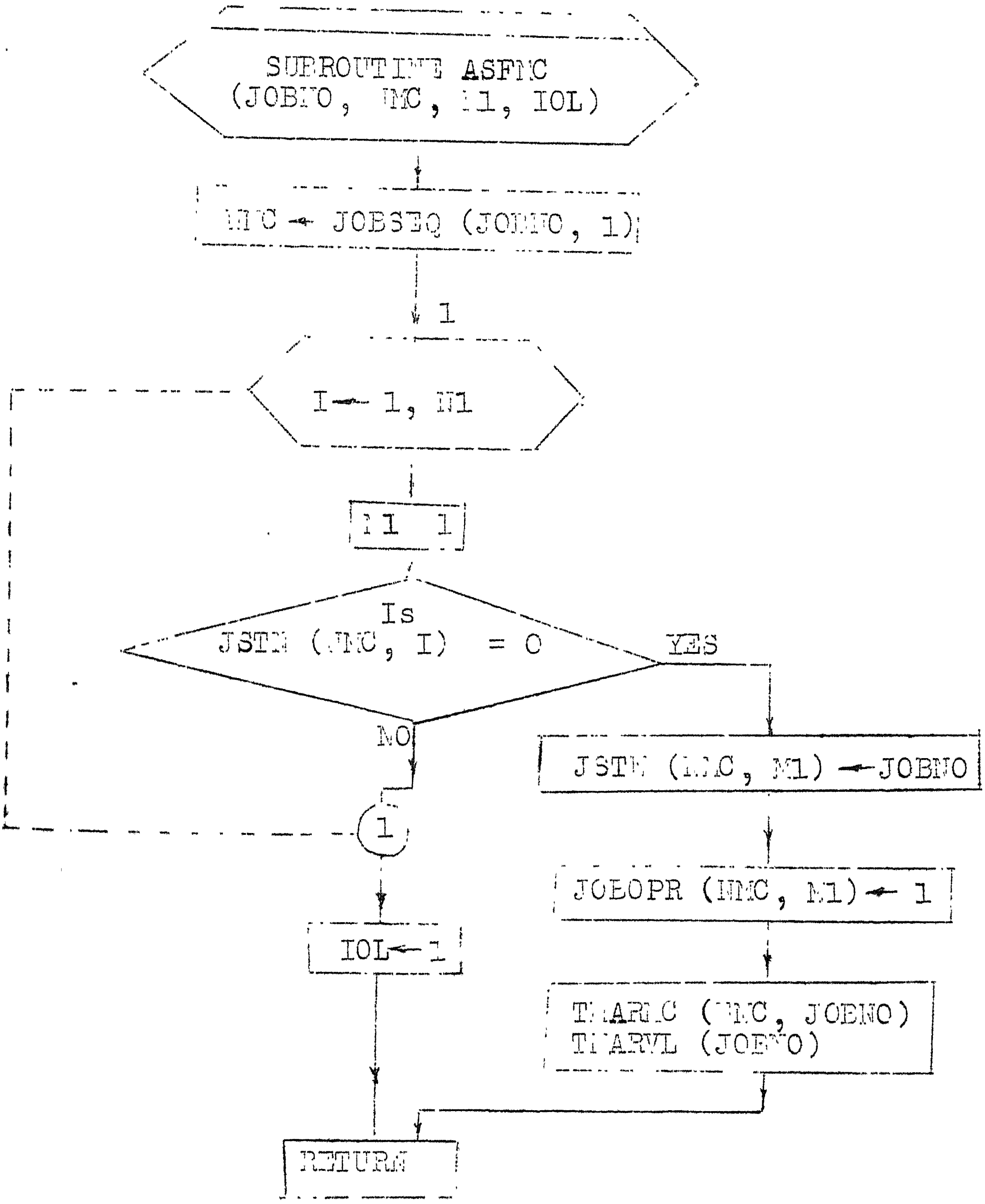


Fig. 7. Flow Chart for ASFMC

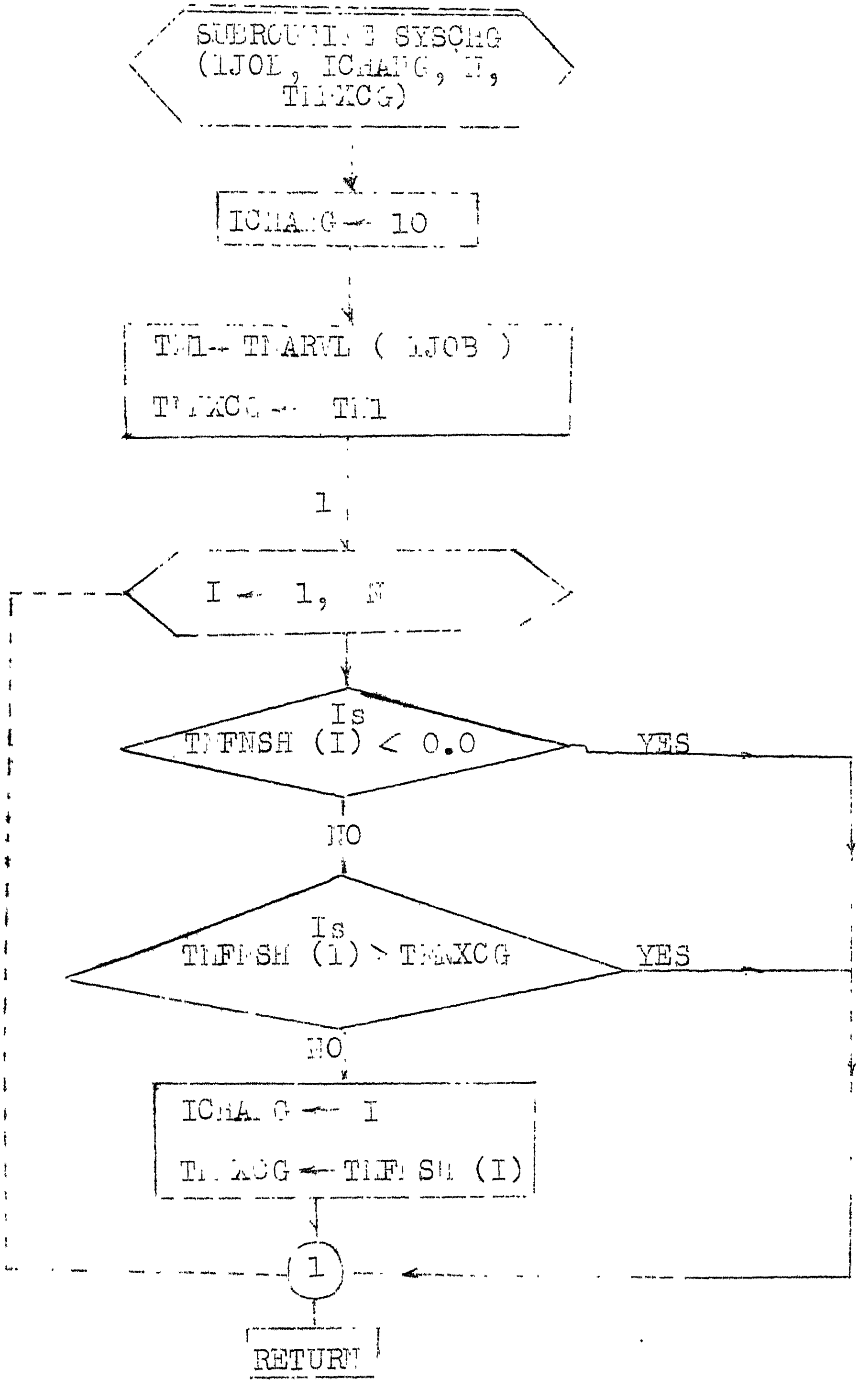


Fig. 8. Flow Chart for SYSCHG

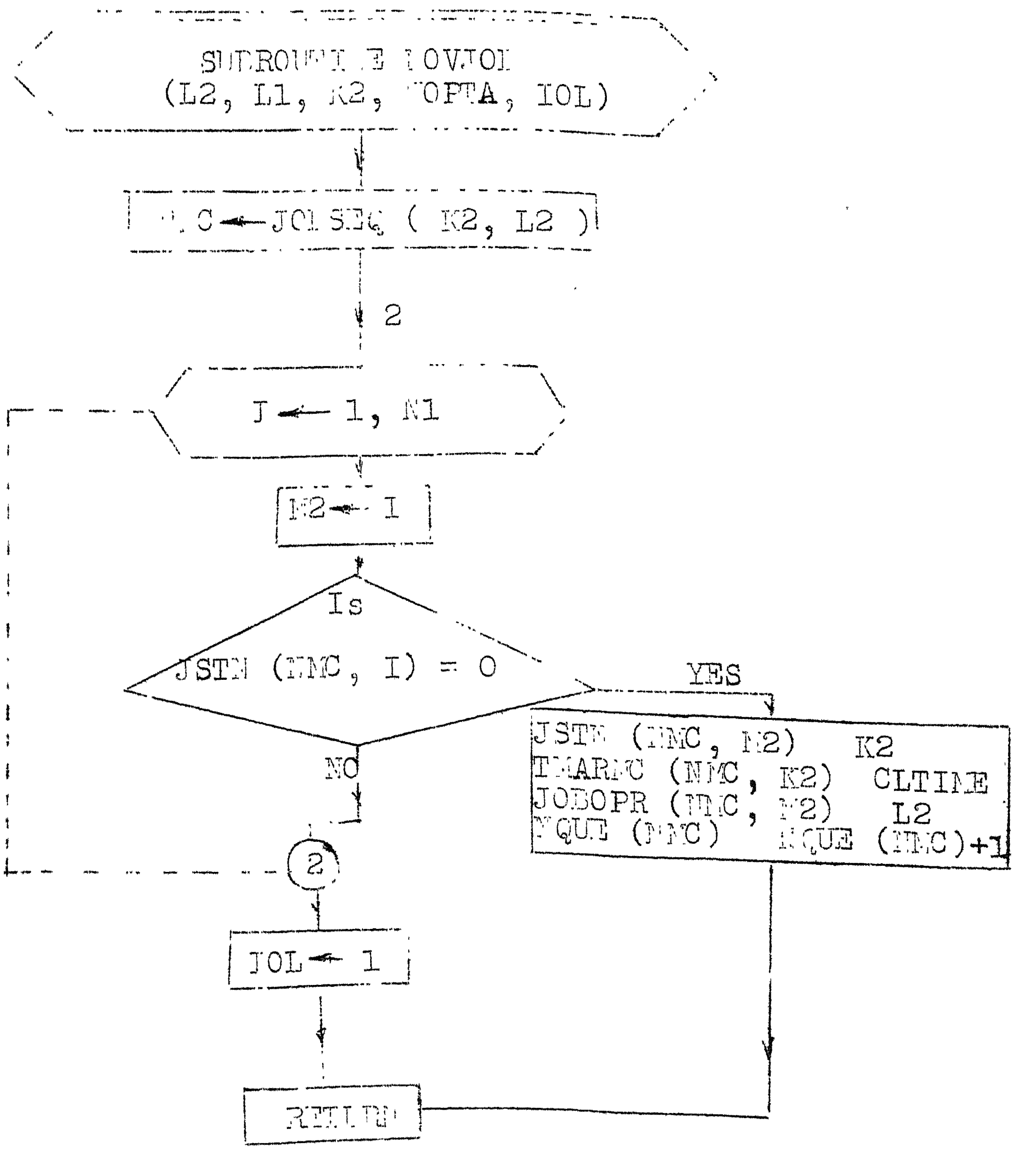


Fig. 9. Flow Chart for MOVJOB

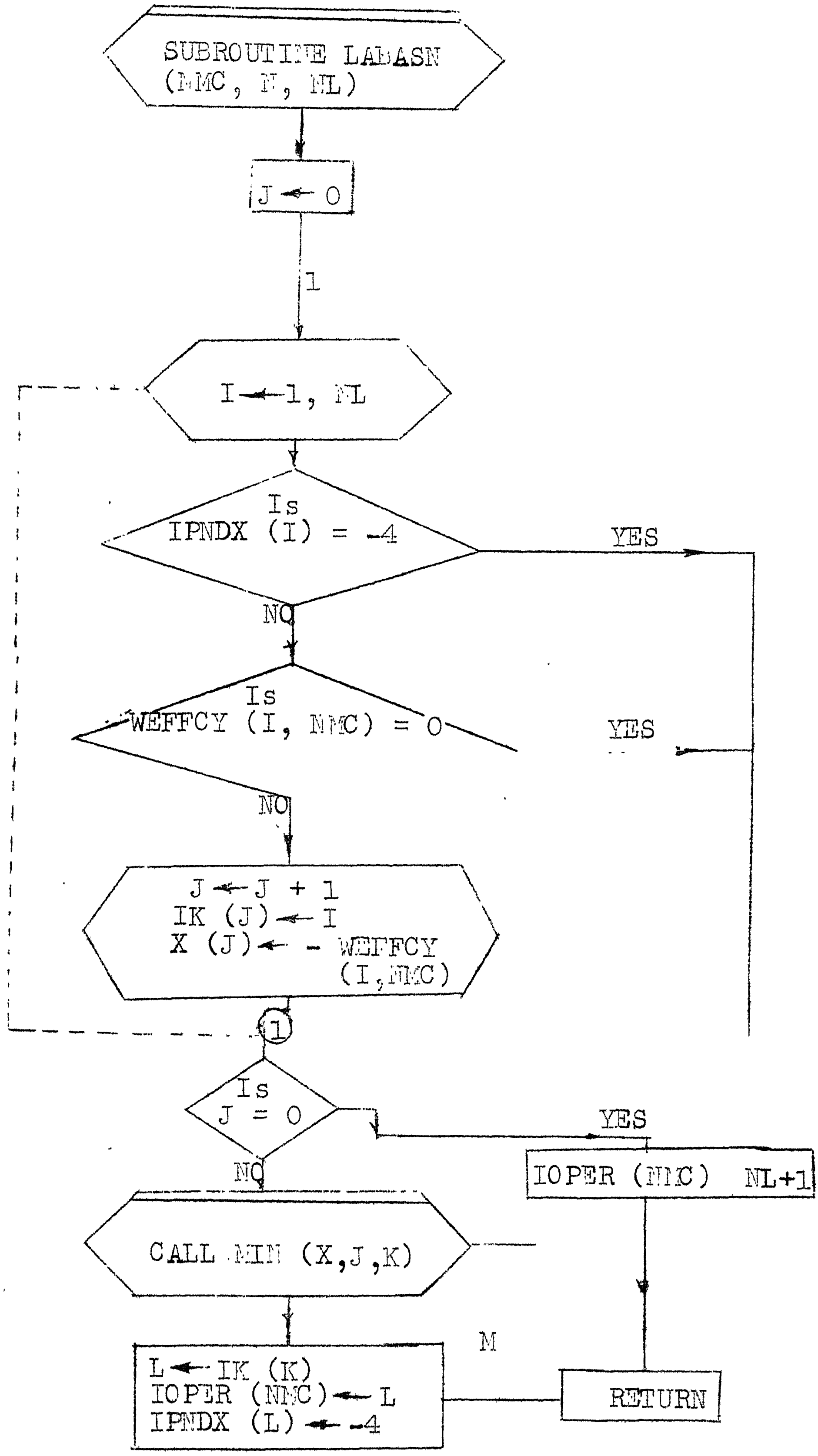


Fig. 10 . Flow Chart for LABASN

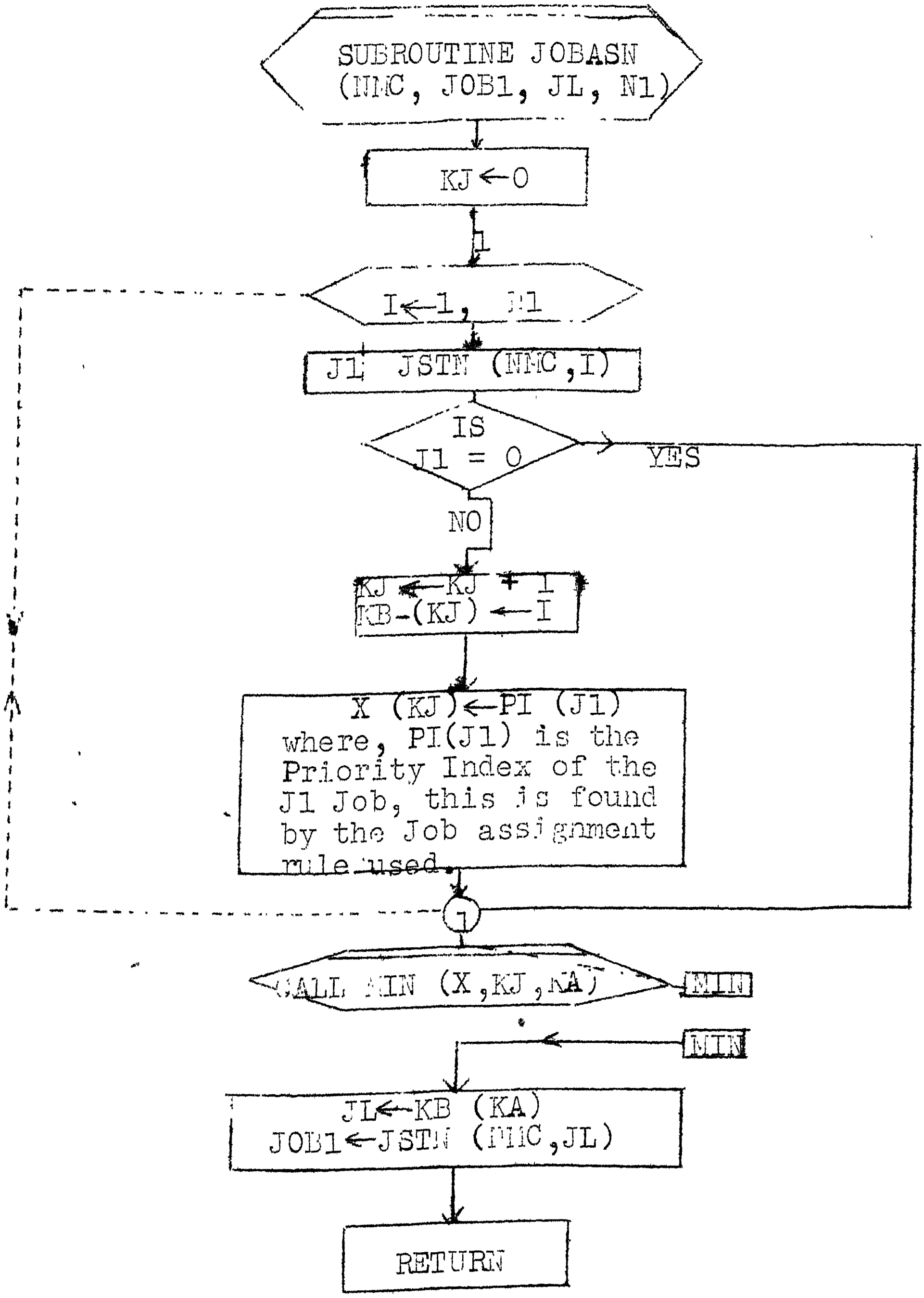


Fig. 11. Flow Chart for JOBASN

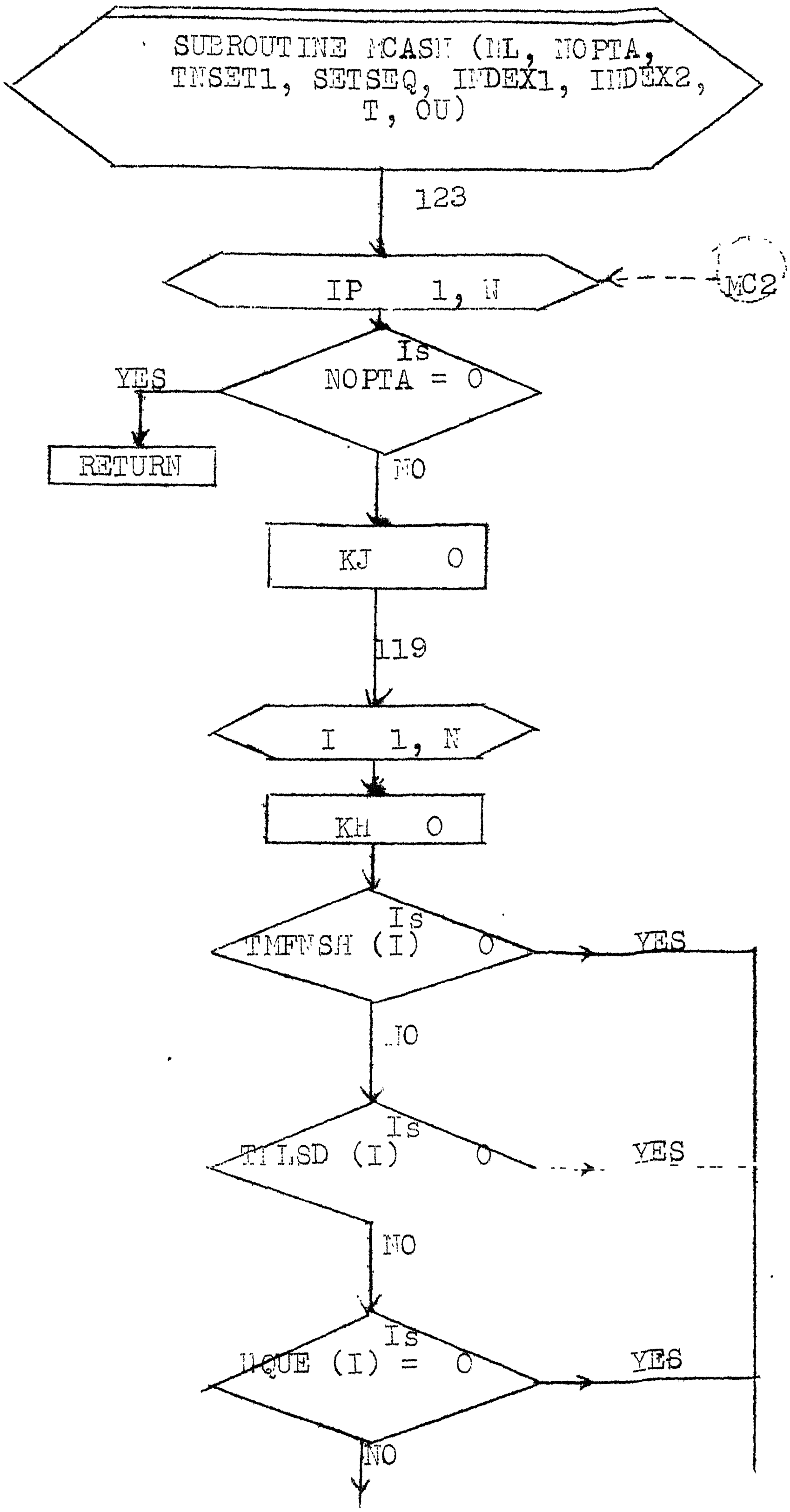


Fig. 12. (Continued)

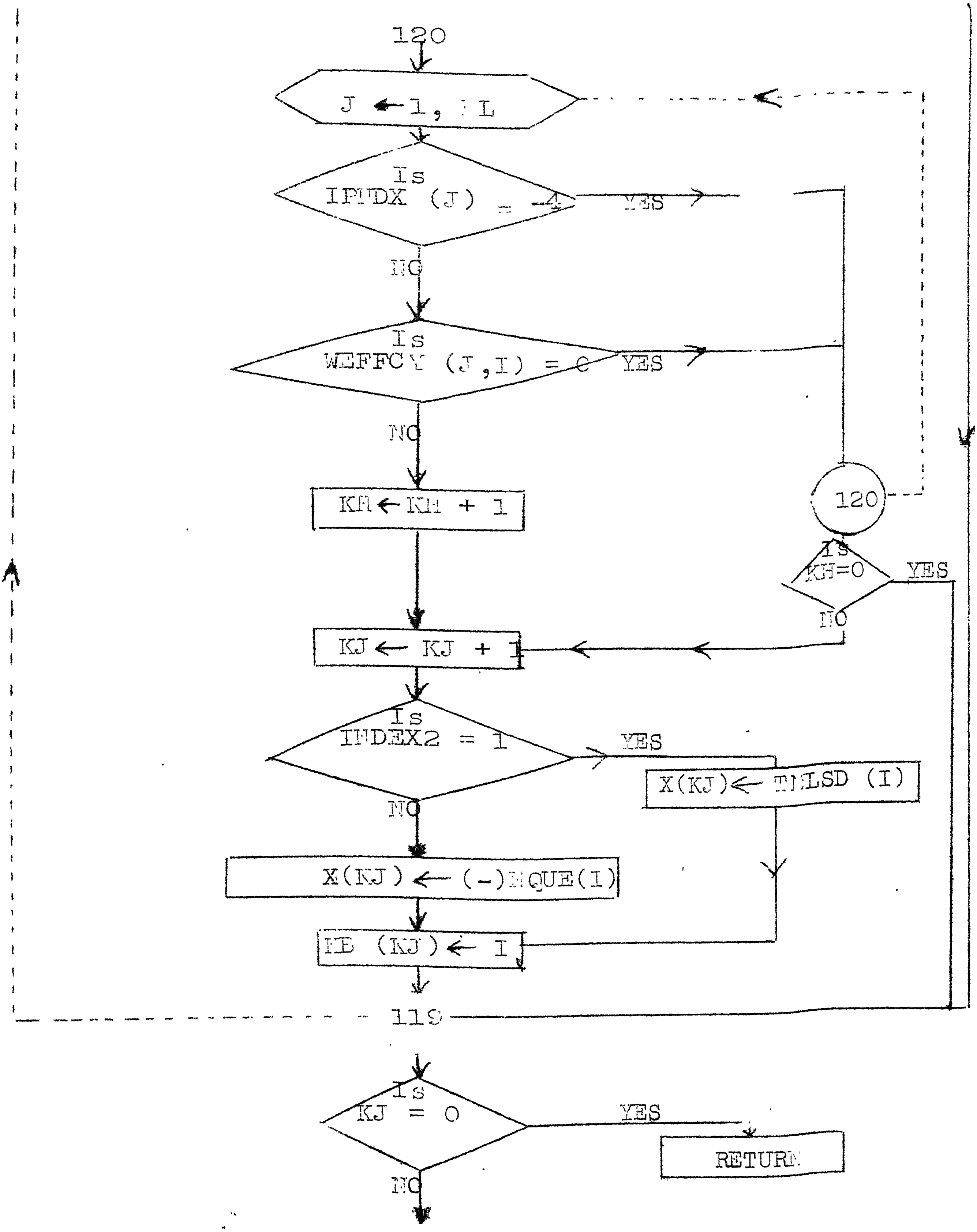


Fig. 12. (Continued)

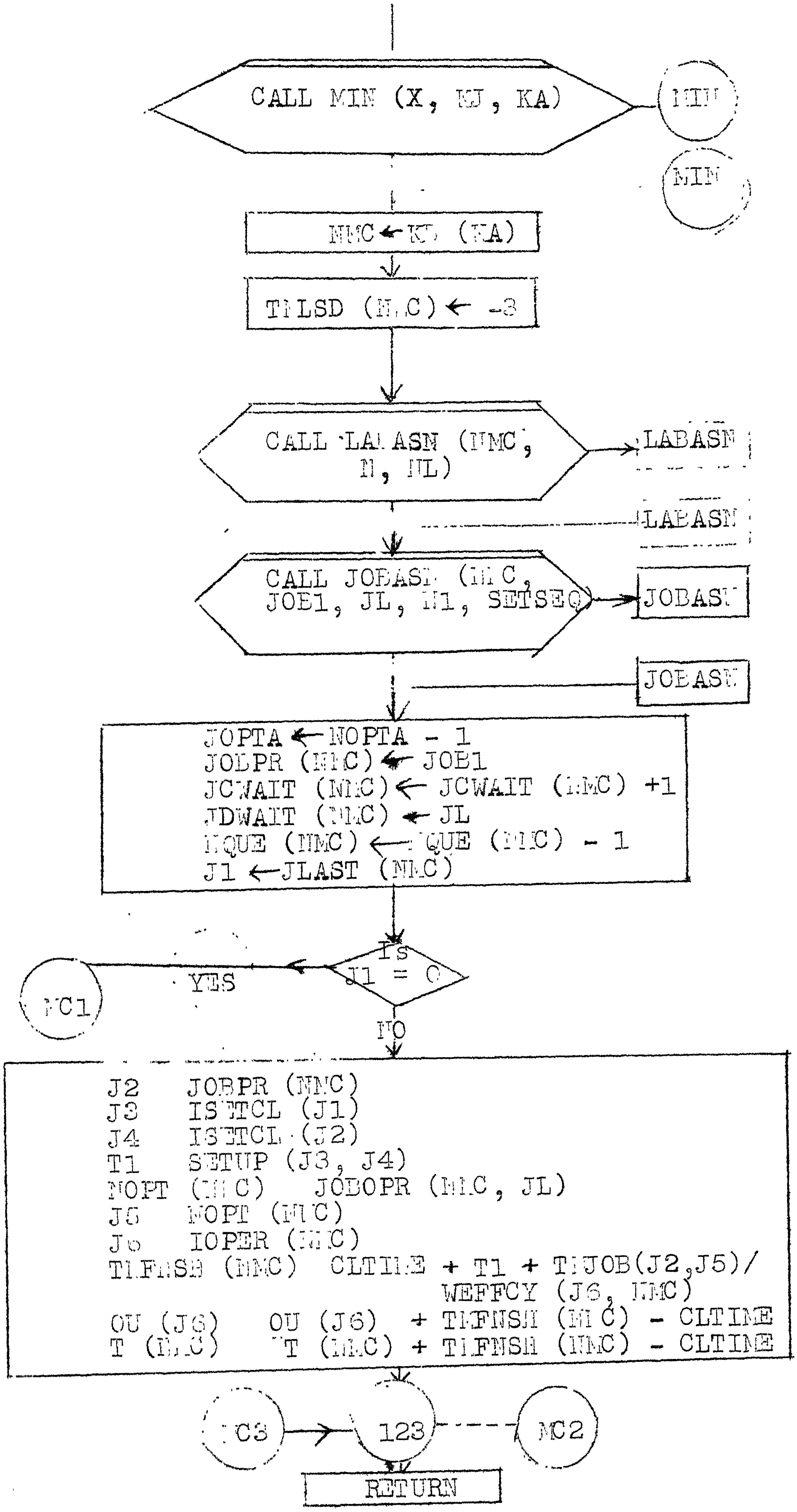


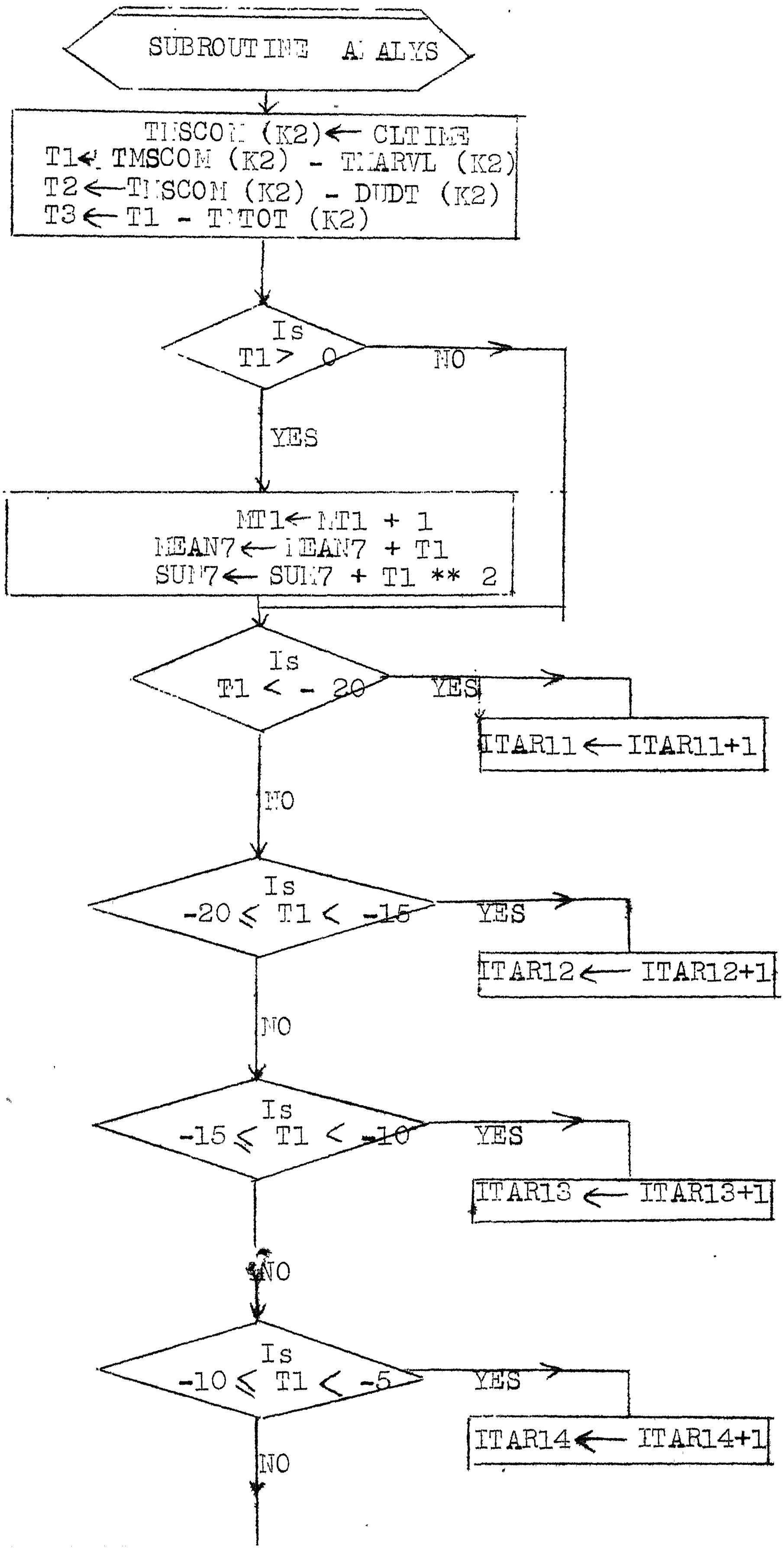
Fig. 12. (Continued)

MC1

```
J2 ← JOBPR (NMC)
J3 ← ISETCL (J2)
J5 ← IOPER (NMC)
NOPT (NMC) ← JOBOPR (NMC, JL)
J6 ← NOPT (NMC)
TMFNSH (NMC) ← CLTIME + TMSET1 (J3) +
               TMJOB (J2, J6) /
               WEFFCY (J5, NMC)
OU (J5) ← OU (J5) + TMFNSH (NMC) - CLTIME
T (NMC) ← T (NMC + TMFNSH (NMC) - CLTIME
```

MC3

Fig. 12. Flow Chart for MCASN


SUBROUTINE ANALYS
TMSCOM (K2) ← CLTIME
T1 ← TMSCOM (K2) - TMARVL (K2)
T2 ← TMSCOM (K2) - DUDT (K2)
T3 ← T1 - TMTOT (K2)
Is T1 > 0
NO
YES
MT1 ← MT1 + 1
MEAN7 ← MEAN7 + T1
SUM7 ← SUM7 + T1 ** 2
Is T1 < - 20
YES
ITAR11 ← ITAR11+1
NO
Is -20 ≤ T1 < -15
YES
ITAR12 ← ITAR12+1
NO
Is -15 ≤ T1 < -10
YES
ITAR13 ← ITAR13+1
NO
Is -10 ≤ T1 < -5
YES
ITAR14 ← ITAR14+1
NO

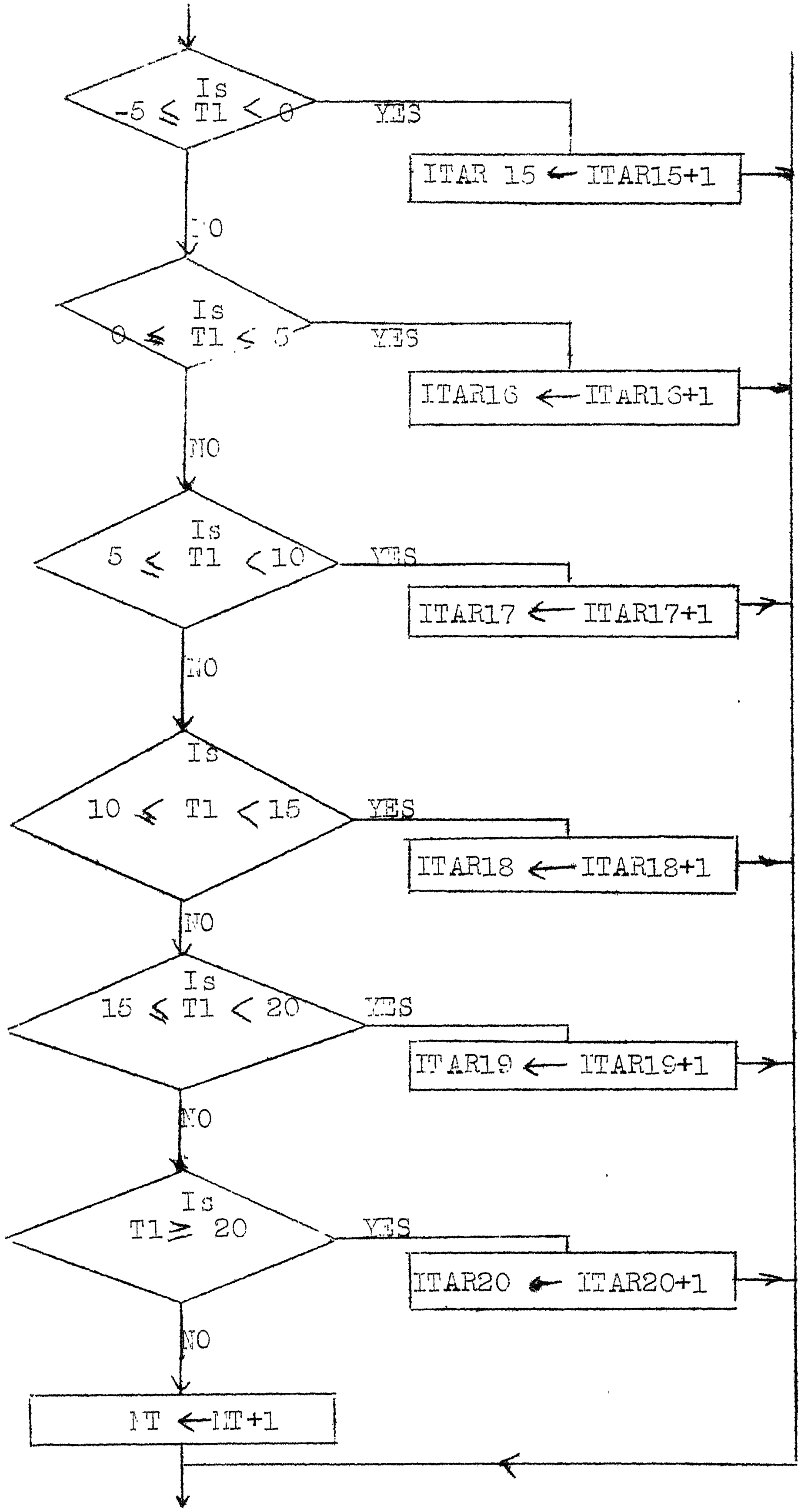


Fig. 13. (Continued)

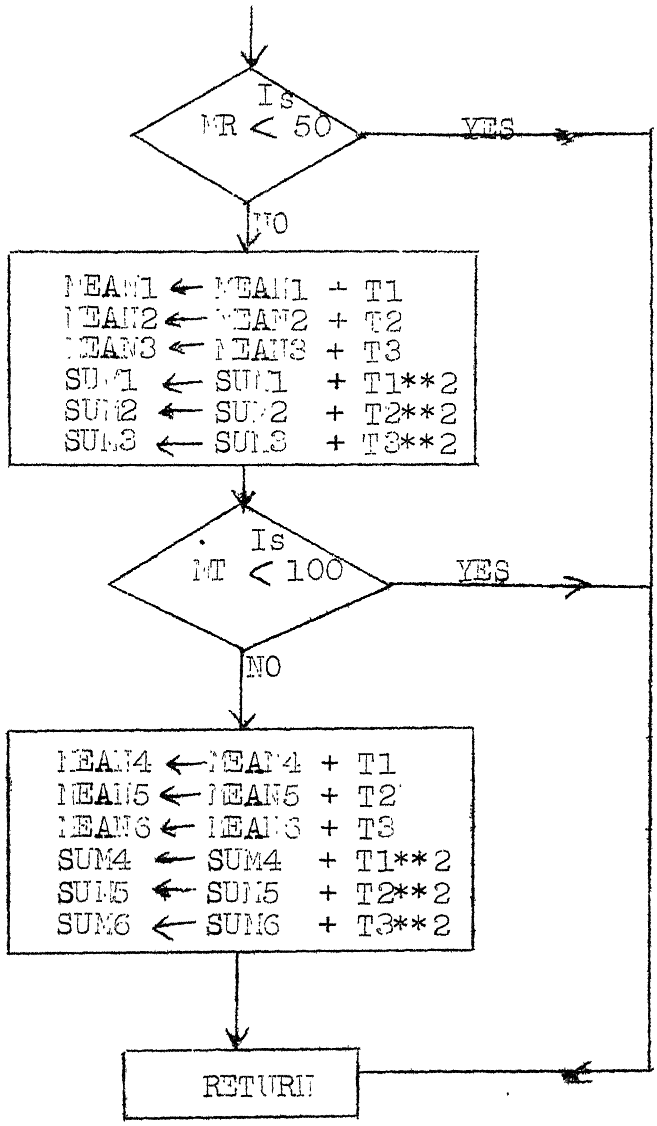


Fig. 13. Flow Chart for ANLYS

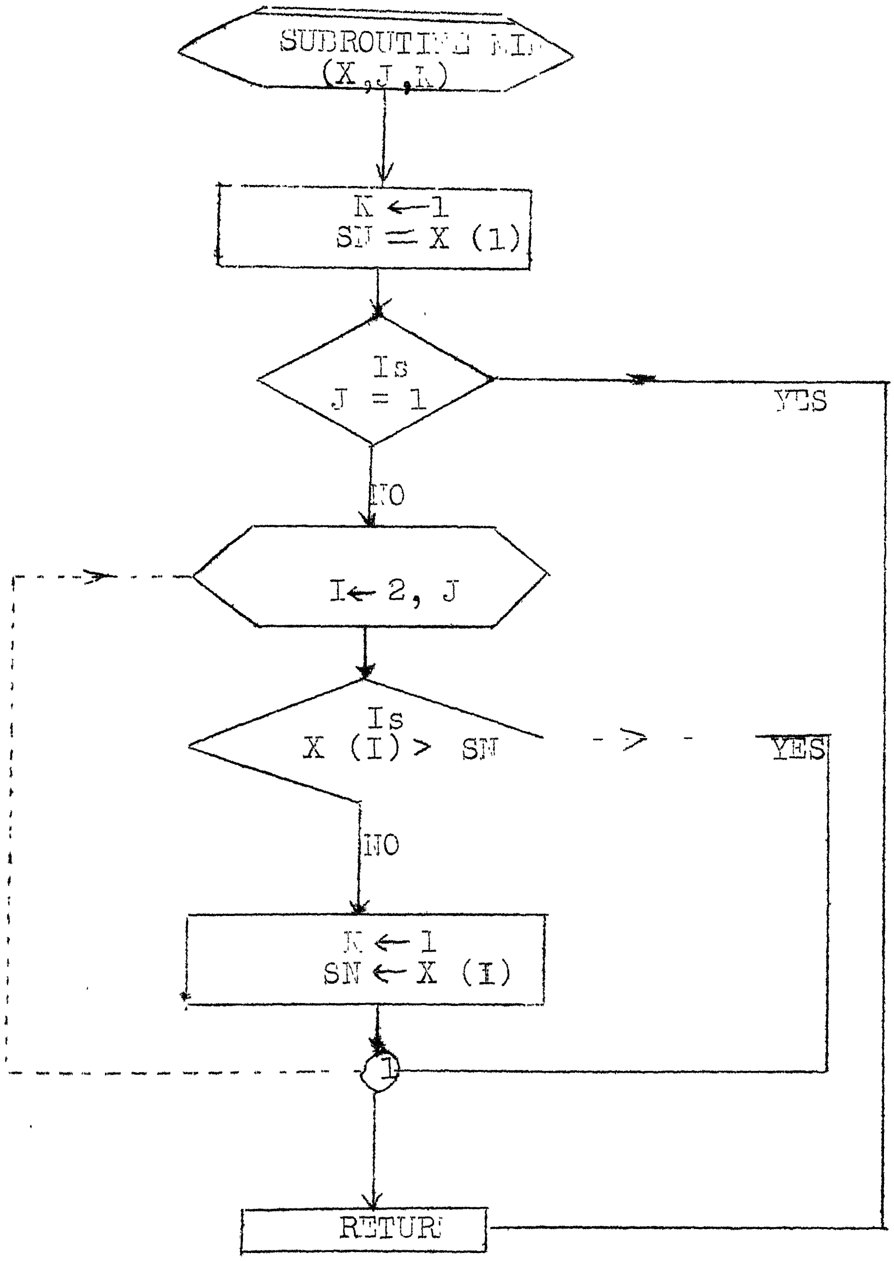


Fig. 14. Flow Chart for MIN

APPENDIX 3

COMPUTER PROGRAM LISTING

FOR

SOLVING TRAVELLING SALESMAN PROBLEM

T R A V E L I N G   S A L E S M A N   P R O B L E M

```
C*****************************************************************************
*                    *                                                       *
*                    *    M A T H   P R O G R A M                            *
*                    *                                                       *
C*****************************************************************************
      DIMENSION JOBSEQ(11)
      REAL LB,LB1,LB2,LB3
      COMMON/OBY/IROW,ICOL,KM
      DIMENSION A(10,10,20),LL(20),B(10,10),S1(10,10),S2(10,10)
      DIMENSION KTM(20)
      DIMENSION IROW(20),ICOL(20)
      READ,N
      KITM=0
      M=1
      L=1
      INDEX=1
      NIS=0
      KM=1
      DO1001I=1,N
      DO1001J=1,N
      A(I,J,M)=0.4*RNDY5(Y5)
      IF(I.EQ.J)A(I,J,M)=100.
 1001 CONTINUE
      PRINT,((A(I,J,M),J=1,N),I=1,N)
      LB=0.
      UB=A(1,2,1)
      N1=N-1
      DO1I=2,N1
 1    UB=UB+A(I,I+1,1)
      UB=UB+A(N,1,1)
      UB1=UB
 13   DO14I=1,N
      DO14J=1,N
 14   B(I,J)=A(I,J,M)
      NIS=NIS+1
      CALL ASSIGN(B,N,LB)
      IF(UB.LE.LB)GOTO102
      IF(UB1.LT.LB)GOTO012
      M1=M
      K1=0
      K2=0
      DO11I=1,N
      DO11J=1,N
      IF(A(I,J,M).EQ.100.)K1=K1+1
      IF(A(I,J,M).EQ.0.)K2=K2+1
   11 CONTINUE
      IF((K1+K2).EQ.(N*N))GOTO12
      CALL REGRET(B,N,IR,IC)
      PRINT,M,IR,IC
```

```
      IROW(KM)=IR
      ICOL(KM)=IC
      KM=KM+1
      CALL PART1(B,N,IR,IC,LB1,S1,LB)
      CALL PART2(B,N,IR,IC,LB2,S2,LB)
      PRINT,LB1,LB2
      IF(LB1.LT.UB)INDEX=INDEX+1
      IF(LB2.LT.UB)INDEX=INDEX+1
      INDEX=INDEX-1
      IF(INDEX.EQ.L)GOTO6
      DO 3I=1,N
      DO4J=1,N
      A(I,J,M1)=S1(I,J)
      A(I,J,INDEX)=S2(I,J)
 4    CONTINUE
 3    CONTINUE
      BL(M1)=LB1
      BL(INDEX)=LB2
      GOTO7
 6    IF(LB1.LT.LB2)GOTO8
      BL(M1)=LB2
      DO 9 I=1,N
      DO 9 J=1,N
 9    A(I,J,M1)=S2(I,J)
      GO TO 7
 8    BL(M1)=LB1
      DO10I=1,N
      DO10J=1,N
 10   A(I,J,M1)=S1(I,J)
 7    CALL MIN(BL,INDEX,M,LB3)
      L=INDEX
      LB=BL(M)
      GO TO 13
 2    PRINT100,UB
 100  FORMAT(5X,*THE OPTIMAL SEQENCE IS 1-2-3-4-5-6----N*,F8.4)
      GOTO15
 12   KITM=KITM+1
      UB1=LB
      KTN(KITM)=M
      DO 30 I=1,INDEX
      DO40J=1,KITM
      IF(I.EQ.KTN(J))GOTO30
 40   CONTINUE
      KN=I
      IF(BL(I).LT.UB1)GOTO31
 30   CONTINUE
      GOTO 45
 31   M=KN
      GO TO 13
 45   DO 20 I=1,N
      PRINT101,(A(I,J,M),J=1,N)
 20   CONTINUE
 101  FORMAT(//5X,6(F20.0))
```

```
      DO1002J=1,N
      IF(A(1,J,M).EQ.0.)GOTO1003
 1002 CONTINUE
 1003 JOBSEQ(1)=1
      JOBSEQ(2)=J1
      K=2
 1006 J2=JOBSEQ(K)
      IF(K.GT.M)GOTO1007
      DO1004J=1,N
      IF(A(J2,J,M).EQ.1.)GOTO1005
      GOTO1004
 1005 K=K+1
      JOBSEQ(K)=J
      GOTO1006
 1004 CONTINUE
 1007 IF(JOBSEQ(1).EQ.JOBSEQ(K))GOTO1008
      PRINT1009
 1009 FORMAT(5X,*THE SEQUENCE IS NOT FEASIBLE*)
      GOTO15
 1008 PRINT1011,(JOBSEQ(I),I=1,K)
 1011 FORMAT(5X,*THE MINIMUM TIME SEQUENCE IS*/5X,11(I3,*--*))
  15   STOP
      END
C**************************************************************************
*                  *
*                  *     S U B R O U T I N E   A S S I G N
*                  *
C**************************************************************************
      SUBROUTINE ASSIGN(D,N,LB)
      REAL LB
      DIMENSION B(10,10),D(10),ROW(10),COL(10)
      DO 1 I=1,N
      DO 2 J=1,N
   2  D(J)=B(I,J)
      CALL  MIN(D,N,KT,RMIN)
      ROW(I)=RMIN
      DO 3 J=1,N
      CA=B(I,J)
      IF(CA.EQ.1000.)GOTO3
      B(I,J)=B(I,J)-RMIN
   3  CONTINUE
1     CONTINUE
      DO4 I=1,N
      DO5J=1,N
   5  D(J)=B(J,I)
      CALL MIN(D,N,KT,CMIN)
      COL(I) =CMIN
      DO6J=1,N
      CA=B(J,I)
      IF(CA.EQ.100.)GOTO6
```

```
      B(J,I)=B(J,I)-CMIN
 6    CONTINUE
 4    CONTINUE
      DO 7 I=1,N
 7    LB=LB+ROW(I)+COL(I)
      RETURN
      END
********************************************************************************
                  *                                                            *
                  *     S U B R O U T I N E   R E G R E T                      *
                  *                                                            *
********************************************************************************
      SUBROUTINE REGRET(B,N,IR,IC)
      COMMON/OPY/IROW,ICOL,KM
      DIMENSION IROW(20),ICOL(20)
      DIMENSION B(10,10),D(10),E(10),X(20),IKR(20),IKC(20)
      KL=0
      DO 1I=1,N
      DO 1J=1,N
      ABC=B(I,J)
      IF(ABC.EQ.0.)GOTO2
      GO TO 1
 2    L=0
      K=0
      IF(KM.EQ.1) GO TO 8
      K1=KM-1
      DO6IJ=1,K1
      IF(I.EQ.IROW(IJ).AND.J.EQ.ICOL(IJ))GOTO1
 6    CONTINUE
 8    DO 3 II=1,N
      IF(II.EQ.J)GOTO3
      L=L+1
      D(L)=B(I,II)
 3    CONTINUE
      DO5II=1,N
      IF(II.EQ.I)GOTO5
      K=K+1
      E(K)=B(II,J)
 5    CONTINUE
      CALL MIN (D,L,KT,RMIN)
      CALL MIN(E,L,KT,CMIN)
      KL=KL+1
      IKR(KL)=I
      IKC(KL)=J
      X(KL)=RMIN+CMIN
 1    CONTINUE
      DO4I=1,KL
 4    X(I)=-X(I)
      CALL MIN(X,KL,MN,RMAX)
      IR=IKR(MN)
      IC=IKC(MN)
      RETURN
      END
```

```
C*****************************************************************************
*                 *                                                          *
*                 *     S U B R O U T I N E   P A R T 1                      *
*                 *                                                          *
C*****************************************************************************
      SUBROUTINE PART1(B,N,IR,IC,LB1,S1,LB)
      REAL LB1,LB
      DIMENSION B(10,10),S1(10,10)
      LB1=LB
      DO1I=1,N
      DO1J=1,N
      IF(I.EQ.IR)GOTO2
      IF(J.EQ.IC)GOTO3
      S1(I,J)=B(I,J)
      GO TO 1
 3    S1(I,J)=100.
      GO TO 1
 2    IF(J.EQ.IC)GO TO 4
      S1(I,J)=100.
      GO TO  1
 4    S1(I,J)=B(I,J)
 1    CONTINUE
      S1(IC,IR)=100.
      CALL ASSIGN(S1,N,LB1)
      RETURN
      END
******************************************************************************
                  *                                                          *
                  *    S U B R O U T I N E   P A R T 2                       *
                  *                                                          *
******************************************************************************
      SUBROUTINE PART2(B,N,IR,IC,LB2,S2,LB)
      REAL LB2,LB
      DIMENSION B(10,10),S2(10,10)
      LB2=LB
      DO 1 I=1,N
      DO 1 J=1,N
      IF(I.EQ.IR.AND.J.EQ.IC)GOTO2
      S2(I,J)=B(I,J)
      GOTO1
2     S2(I,J)=100.
1     CONTINUE
      CALL ASSIGN(S2,N,LB2)
      RETURN
      END
******************************************************************************
                  *                                                          *
                  *    S U B R O U T I N E   M I N                           *
                  *                                                          *
******************************************************************************
      SUBROUTINE MIN(P,N,K,RMIN)
      DIMENSION P(10)
      K=1
      SUM=P(1)
```

```
      IF(N.EQ.1)GOTO2
      DO1I=2,N
      IF(P(I).GT.SUM)GOTO1
      K=I
      SUM=P(I)
 1    CONTINUE
      RMIN=SUM
      RETURN
      END
$ENTRY
```